

**Oracle® ODI Planogram Import**

User Guide, Volume 1

Release 14.1

**E59347-01**

December 2014

Copyright © 2014, Oracle and/or its affiliates. All rights reserved.

Primary Author: Phil Wells

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

#### **Value-Added Reseller (VAR) Language**

##### **Oracle Retail VAR Applications**

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

- (i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.
- (iii) the software component known as **Access Via**<sup>™</sup> licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.
- (iv) the software component known as **Adobe Flex**<sup>™</sup> licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all

reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.





---

---

# Contents

<b>Send Us Your Comments</b> .....	xi
<b>Preface</b> .....	xiii
<b>1 Introduction to Target Bridge Documentation</b>	
Chapters .....	1-1
Other Sections .....	1-2
<b>2 Introduction to ODI Planogram Import</b>	
<b>Executive Overview</b> .....	2-1
Source Bridge .....	2-1
Target Bridge .....	2-2
<b>Audience and Skill Set</b> .....	2-2
Implementers .....	2-2
Installers .....	2-2
MSP Administrators .....	2-2
Oracle Data Integrator .....	2-3
<b>Deployment Diagram</b> .....	2-3
Source Bridge .....	2-3
Target Bridge .....	2-4
Latency .....	2-4
<b>Planogram Import Process Overview</b> .....	2-5
<b>Size and Capacity Requirements</b> .....	2-6
<b>3 Overview of Planograms</b>	
<b>Overview of Planograms</b> .....	3-1
Purpose of Planograms .....	3-1
Planogram Terminology .....	3-1
<b>How Planogram Data is Stored</b> .....	3-3
Planogram Hierarchy .....	3-4
Planogram Components .....	3-4
Planogram Information .....	3-5
Planogram Data .....	3-6

## 4 Overview of Oracle Data Integrator

<b>Overview of Data Transfer Methods</b> .....	4-1
<b>ODI Architecture</b> .....	4-2
ODI Studio (Desktop Option) .....	4-2
Navigators.....	4-3
Local Agent.....	4-3
Repositories.....	4-3
Master Repository.....	4-4
Work Repositories.....	4-4
Development Repositories .....	4-4
Execution Repositories.....	4-4
Relationship Between Repositories .....	4-4
Agents .....	4-5
Methods of Executing Data Transfers using Agents .....	4-5
Scheduling Agent.....	4-6

## 5 Overview of Installing the Target Bridge

<b>Database Instance</b> .....	5-2
<b>Files</b> .....	5-3
<b>Macro Space Planning</b> .....	5-3
Ensure Base Data is Correct.....	5-3
Configure Planogram Import Specific Information .....	5-4
<b>Install ODI Studio</b> .....	5-4
<b>Install Agents</b> .....	5-5
<b>Configure ODI</b> .....	5-5
Create Access to the ODI Staging Area.....	5-5
Create/Import the ODI Master Repository.....	5-5
Create the ODI Work Repository.....	5-6
<b>Configure the Agents</b> .....	5-6

## 6 Configuring the Database Instance

<b>Configuring the Database Instance</b> .....	6-1
Creating the Schemas required by ODI .....	6-1
Creating the Schema for the ODI Master Repository .....	6-2
Creating the Schema for the ODI Work Repository .....	6-2
Creating the Schema for the ODI Staging Area.....	6-3
Running the Upgrade Script Against the MSP Database.....	6-3
Running the Upgrade Script .....	6-4

## 7 Creating the Data Folders

<b>Creating the Required Data Folders</b> .....	7-1
Creating the folders for Import and Archived XML files .....	7-1
Creating the Folders .....	7-1
Creating the Folder for and Placing the XSD File .....	7-2

## 8 Configuring the MSM Administration Module

Installation Permissions.....	8-1
Ensure Base Data is Correct.....	8-1
Administration Module .....	8-2
Fixture Studio .....	8-2
Product Studio .....	8-3
General Information .....	8-3
Configuring ODI Planogram Import Specific Information.....	8-3
Configuring Planogram Import Styles.....	8-3
Stretchable Blocks .....	8-5
Configuring the System Variables .....	8-6
Accessing and Using the System Variable Dialog Box.....	8-6
Configuring System Variables for the Import and Archive Directories .....	8-6
Configuring System Variables for the Planogram Import Defaults .....	8-7
Configuring System Variables for the Default Blocks .....	8-7
Stretchable Blocks .....	8-9

## 9 Installing ODI Studio

Installing ODI Studio in a Desktop Environment .....	9-1
Reinstalling ODI and Connecting to Existing Repositories .....	9-7

## 10 Installing ODI Agents

Creating the Agent .....	10-1
MSMODIAgent .....	10-10
Starting and Testing the Agent.....	10-10
Starting the Standalone Agent .....	10-11
Testing the Agent from ODI Studio .....	10-12
ODI Agents and Scheduling .....	10-13

## 11 Configuring ODI Studio

Configuring Access to the ODI Staging Area.....	11-1
Encoding the Password.....	11-2
Create and Configure ODI Master Repository.....	11-3
Create Master Repository.....	11-3
Creating a Login for the Master Repository .....	11-5
Configure Topology - Physical Architecture .....	11-7
Target Database.....	11-7
XML Technology.....	11-9
Verify Topology - Logical Architecture .....	11-12
Creating the Work Repository .....	11-13
Creating the Work Repository .....	11-13
Creating the Login for the Work Repository .....	11-16
Creating the Work Repository Login Later.....	11-16
Creating the Work Repository Login.....	11-17
Importing the Scenarios .....	11-18

Reinstalling ODI and Connecting to Existing Repositories .....	11-21
--	-------

## 12 Configuring ODI Agents

<b>Installation Permissions</b> .....	12-1
Deployment Details .....	12-1
<b>MSMODI Agent</b> .....	12-1
<b>Encoding the Passwords</b> .....	12-2
<b>Configuring Access to the ODI Staging Area</b> .....	12-2
Encoding the Password.....	12-4
<b>Configuring the odiparams Batch File</b> .....	12-4
Mandatory Parameters.....	12-5
Parameters for Agent Startup Program .....	12-6
Parameters for ODI Tool User Credentials .....	12-6
Parameter Required by Startcmd, Startscen and Restartsession.....	12-7
Parameters Required to Tune ODI Configuration .....	12-7
<b>Configuring the Physical and Logical Agents in ODI Studio</b> .....	12-9
Creating the Physical Agent .....	12-9
Creating the Logical Agent.....	12-10
<b>Starting and Testing the Agent</b> .....	12-11
Starting the Agent .....	12-11
Testing the Agent from ODI Studio .....	12-12
<b>ODI and Scheduling</b> .....	12-14

## 13 Running ODI Planogram Imports

<b>Overview</b> .....	13-1
Forms of Planogram Import .....	13-1
Running Planogram Imports.....	13-1
<b>Manually Running a Planogram Import from ODI Studio</b> .....	13-2
Preparatory Actions.....	13-2
Running the Planogram Import.....	13-2
Connect to the Work Repository .....	13-2
Starting a Scenario from the Operator Navigator .....	13-3
<b>Running using ODIs Scheduling Tool and an Agent</b> .....	13-6
Preparatory Actions.....	13-6
Setting up the Schedule.....	13-7
Connect to the Work Repository in ODI Studio.....	13-7
Specifying the Schedule in the Designer or Operator Navigator.....	13-8
Starting and Stopping the Standalone Agent.....	13-12
Starting the Standalone Agent .....	13-12
Other Command Line Options .....	13-13
Stopping the Standalone Agent .....	13-13
<b>Running Scheduled Tasks using Batch Files</b> .....	13-15
Preparatory Actions.....	13-15
Setting up the Schedule.....	13-15
Connect to the Work Repository in ODI Studio.....	13-15
Specifying the Schedule in the Designer or Operator Navigator.....	13-16
Controlling the Import Schedule using Batch Files .....	13-19

Batch File to Start the Standalone Agent .....	13-19
Batch File to Stop the Standalone Agent .....	13-20
Scheduling the Standalone Agent .....	13-21
<b>14 Overview of Planogram Import Process</b>	
User Determined Parameters Affecting Import Process.....	14-1
Operating Logic .....	14-2
<b>15 Using Planogram Import Logs</b>	
Overview of using Import Logs.....	15-1
<b>ODI Log Information</b> .....	15-1
Accessing the ODI Log information .....	15-1
Log Information Results.....	15-3
Variables and Log Results .....	15-3
Error Result on End of Run .....	15-5
<b>Macro Space Planning Log Information</b> .....	15-6
AVTTB_ODI_LOG Table .....	15-7
Accessing Data from the AVTTB_ODI_LOG Table.....	15-7
Purging the AVTTB_ODI_LOG Table .....	15-7
SNP_CHECK_TAB Table.....	15-8
Accessing Data from the SNP_CHECK_TAB Table .....	15-9
Purging the SNP_CHECK_TAB Table.....	15-9
Collation of Log Information.....	15-9
<b>16 Troubleshooting Initial Problems</b>	
<b>General Issues</b> .....	16-1
Planogram Import does not currently handle placeholders.....	16-1
<b>Installing and Configuring ODI</b> .....	16-2
Dynamic and Static IP Addresses.....	16-2
Unable to Connect to Repository .....	16-2
Repository Already Exists Warning when Creating Master Repository .....	16-2
Old Login Information Exists After Reinstalling ODI.....	16-3
<b>ODI Studio Functionality</b> .....	16-4
Navigator disappears from View .....	16-4
<b>Running an Import</b> .....	16-4
Location of Import and Archive Folders .....	16-4
ODI Staging Area Schema Name Not Exactly Seven Characters .....	16-5
Lack of Memory Error (Java Heap Size Too Big) .....	16-5
Path to XSD file must be specified in full for XML configuration .....	16-5
<b>A Appendix: Installation Detail Example</b>	
Appendix: Installation Detail Example .....	A-1
<b>B Appendix: Installation Detail Checklist</b>	
Appendix: Installation Detail Checklist.....	B-1

**C Appendix: Mandatory XML Elements**

Appendix - Mandatory XML Elements ..... C-1

**Glossary**

---

---

## Send Us Your Comments

Oracle Retail ODI Planogram Import, Volume 1, Release 14.1

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

---

---

**Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

---

---

Send your comments to us using the electronic mail address: [retail-doc\\_us@oracle.com](mailto:retail-doc_us@oracle.com)

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.





---

---

# Preface

This document describes how to import planograms into the Macro Space Planning database using the Target Bridge part of the ODI Planogram Import solution.

## Audience

This document is intended for Implementers, System Administrators and Users.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

For more information, see the following documents in the Oracle Retail Macro Space Planning 14.1 documentation set, the Oracle Retail Macro Space Management Release 14.1 documentation set or in the Oracle Data Integrator 11.1.1.7 documentation set:

- *Oracle Retail Macro Space Management Release Notes*
- *Oracle Retail Macro Space Planning Installation Guide*
- *Oracle Retail Macro Space Management Administration User Guide*

For more information about ODI, see the following documents:

- *Oracle Data Integrator Installation Guide*
- *Oracle Data Integrator Developers Guide*

Additional Oracle Data Integrator guides may be found at:

<http://www.oracle.com/technetwork/middleware/data-integration/documentation/index.html>

## Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

## Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.3) or a later patch release (for example, 13.3.1). If you are installing the base release or additional patches, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

## Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:

[http://www.oracle.com/technology/documentation/oracle\\_retail.html](http://www.oracle.com/technology/documentation/oracle_retail.html)

(Data Model documents are not available through Oracle Technology Network. These documents are packaged with released code, or you can obtain them through My Oracle Support.)

Documentation should be available on this Web site within a month after a product release.

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---

---

# Introduction to Target Bridge Documentation

This brief section gives an overview to the Target Bridge documentation. It is intended to allow people who do not need to read the entire document to identify which sections are pertinent to the information they are seeking.

---

---

**Note:** This document is part of the Macro Space Planning (MSP) documentation set. Screen captures and steps associated with the Oracle technologies (such as ODI) that MSP leverages may vary as those technologies are further developed by Oracle. The screen shots and steps in this guide are provided for reference purposes only (and for the assistance of MSP partners and retailers) and are not meant to definitively represent ODI functionality. For the latest information about ODI, see ODIs current documentation set.

---

---

The sections in this guide are structured as follows:

## Chapters

### **Introduction to ODI Planogram Import**

This brief chapter gives an overview of how planogram import works using Oracle Data Integrator (ODI).

### **Overview of Planograms**

This brief chapter is designed for system implementers or administrators who are not familiar with the concept of planograms. It contains information on what a planogram is, together with how planogram information is stored in the Macro Space Planning (MSP) database.

### **Overview of Oracle Data Integrator**

This brief chapter is designed for system implementers or administrators who are not familiar with ODI. It explains the basic concepts within ODI.

### **Overview of Installing the Target Bridge**

This chapter summarizes the basic stages in installing the Target Bridge. Read this chapter prior to any installation task as it lists the tasks and the sequence they should be performed.

### **Configuring the Database Instance**

This chapter explains how to create the additional schemas required in the Database Instance.

### **Creating the Data Folders**

Folders are required to hold the XML files for import, the archived XML files and the XSD file used during planogram import. This chapter details how to create them.

### **Configuring the MSM Administration Module**

This chapter describes settings required within the MSM Administration Module. This information will be referenced during planogram import.

### **Installing ODI Studio**

This chapter describes how to install ODI Studio. ODI Studio is used for specifying the details required for planogram imports.

### **Installing ODI Agents**

The ODI agents are used to run scheduled planogram imports. Agents may be Local (installed on the same machine as ODI Studio) or Standalone (installed on a different machine to ODI Studio). This chapter explains how to install both.

### **Configuring ODI Studio**

This chapter describes how to configure ODI Studio.

### **Configuring ODI Agents**

This chapter explains how to configure both local and standalone agents.

### **Running ODI Planogram Imports**

This chapter explains how to execute planogram imports using the target bridge. They may be initiated manually from ODI Studio, via the command line or using batch files. This section explains all three methods.

### **Overview of Planogram Import Process**

This section gives an overview of the operations carried out during the planogram import process.

### **Using Planogram Import Logs**

This section describes how information is stored in the planogram import logs.

### **Troubleshooting Initial Problems**

This chapter explains how to troubleshoot some of the problems likely to be encountered when first installing the ODI Planogram Import solution.

## **Other Sections**

### **Appendix: Installation Detail Example**

This appendix contains a filled in checklist with the data used for an example installation of the ODI Planogram Import solution.

**Appendix: Installation Detail Checklist**

This appendix contains a table useful for recording passwords, user names and other information that may be required during the installation and configuration process.

**Appendix - Mandatory XML Elements**

This appendix contains a list of the elements that are mandatory in the XML file. It is intended for quick reference purposes.

**Glossary**

This glossary explains some of the terms that may be met while installing, configuring and operating the ODI Planogram Import solution.



---

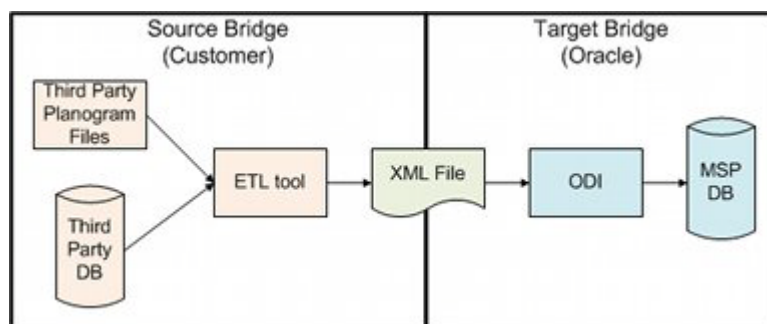
## Introduction to ODI Planogram Import

This chapter of the Target Bridge documentation gives an executive overview of the Planogram Import solution.

### Executive Overview

This section gives a brief description of the planogram import solution. It is in two parts: one the responsibility of the customer and one the responsibility of Oracle. Once implemented, it will allow planograms from a third party planogram design tool to be imported into the Macro Space Planning (MSP) application and used for Space Planning purposes.

**Figure 2–1 Solution Overview**



### Source Bridge

The Source Bridge is the responsibility of the customer.

You will need to configure an Extract, Transform and Load (ETL) tool to take data from a third party planogram design tool (stored in either a database or in planogram files) and convert it into XML format. There may be one or multiple XML files:

- All planograms could be exported to a single XML file.
- If there is more than one planogram design software application in use, each could result in a separate XML file.
- It is possible to generate one XML file per planogram - although this is not as efficient an import method as a single XML file.

This XML information will then be exported to a defined location which serves as the import source for the Target Bridge.

## Target Bridge

The Target Bridge is the responsibility of Oracle.

The starting point is the XML files generated by the Source Bridge. The Target Bridge is capable of importing data from multiple XML files in a single data import. Once the import is initiated, Oracle Data Importer (ODI) then extracts data from the XML file(s), transforms it into the format required by the MSP database and inserts it into that database.

Once imported into the MSP database, the planograms are available to both Macro Space Management (MSM) or In-Store Space Collaboration (ISSC) - the two applications that make up the Macro Space Planning suite.

## Audience and Skill Set

The implementation, installation and operation of the Planogram Import Solution will probably be carried out by three distinct groups of people. These are:

- Implementers
- Installers
- MSP Administrators

### Implementers

Implementers should understand how to generate the XML file that is the output from the source bridge. In order to achieve this they should have:

- A good knowledge of the ETL tool that is used to generate the XML file.
- The ability to generate an XML file matching the requirements specified by Oracle.
- The ability to help MSP administrators troubleshoot problems.

### Installers

Installers are responsible for making updates to the MSP schema, creating additional schemas in the database instance holding the MSP schema, installing ODI and configuring it when installed. In order to do this installer must have:

- A detailed knowledge of Oracle databases.
- Sufficient privileges to create users, roles and schemas in the required databases.
- The ability to install ODI.
- The ability to configure ODI to the supplied instructions.

### MSP Administrators

MSP administrators should have an in depth knowledge of how MSP is configured. Specifically, they should understand:

- How to configure options in the Administration module.
- How to configure blocks in Fixture Studio.
- The concept of planograms.
- How planograms are stored in the MSP database.
- How to place planograms in MSM and ISSC.



They are expected to do the following:

- Configure MSP in preparation for executing a planogram import.
- Assist in scheduling planogram imports.
- Examine the planogram import logs to identify in any failures have occurred.
- Assist in identifying the causes of errors in planogram imports.

## Oracle Data Integrator

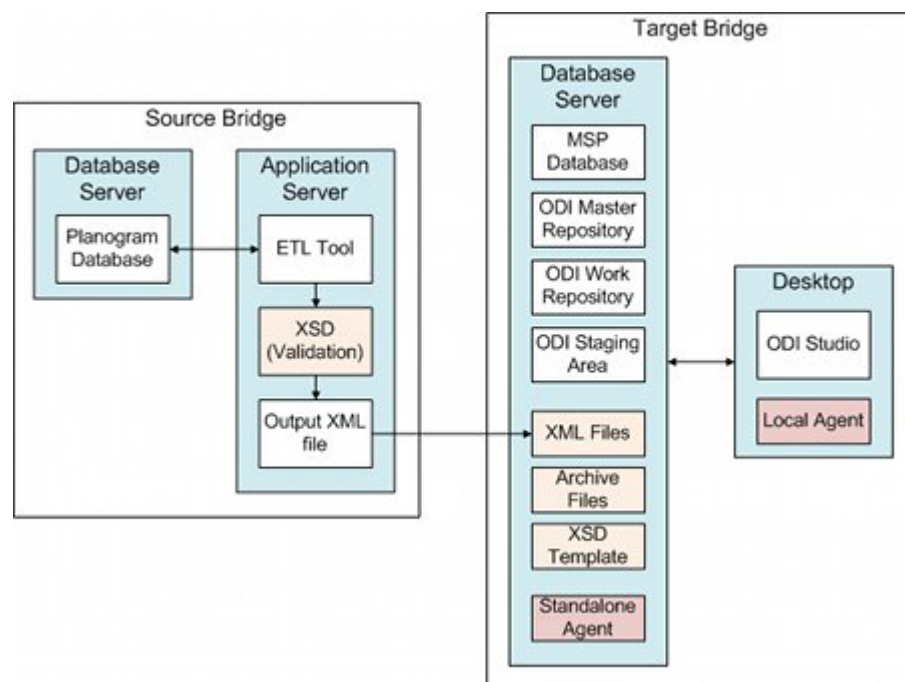
Personnel involved in using ODI should have a basic understanding of the application including:

- ODI terminology.
- Use of main ODI functionality.
- Reading ODI log information.

## Deployment Diagram

This section contains a generalized deployment diagram for ODI planogram import deployed in a Windows environment. If ODI is deployed in an application server (WebLogic) environment, it requires a modified approach - not currently covered in this document.

**Figure 2–2** *Deployment in a Windows Environment*



### Source Bridge

The source bridge (which is the client's responsibility) may contain:

### **Database Server**

This would contain the database holding the planogram information.

### **Application Server**

This would contain the client's ETL tool. This would generate an XML file containing the required planogram information. It would then be validated against an XSD. On completing validation of the XML file it would be written/transferred to the database server in the Target Bridge.

## **Target Bridge**

The target bridge (configured by Oracle) may contain:

### **Database Server**

The database server contains a database instance with:

- **MSP Database:** this holds the planogram tables that are the target for the import, together with information referenced during the import, information on the directories used and log tables holding the results of the import. It also has temporary tables used to hold data during the planogram import.
- **ODI Master and Work Repositories:** these hold the configuration information used by ODI to execute the planogram import.
- **ODI Staging Area:** this is a schema used as a temporary holding space for planogram data while it is being processed by ODI and before it is imported into the MSP database.

The database server also contains a series of directories and files:

- **XML File:** this is transferred from the source bridge and contains the information on the planograms to be imported.
- **Archive Files:** these are XML files that have been imported and are being held for reference purposes.
- **XSD Template:** this holds information on the structure of the XML file. It is used to create temporary tables in the Staging Area. These temporary tables are used to validate the data prior to transformation and import into the MSP database.

The database server may also hold a copy of a standalone agent. This agent is invoked at specific times (either by using ODI's own scheduling tool or via an external scheduling tool). Once invoked it calls the varying actions required to execute a data transfer.

### **Desktop**

This holds a copy of ODI Studio. ODI Studio is used to configure details of the planogram import prior to it being executed at run time. It may also hold a copy of a run time agent (local agent) that can be used to execute scheduled imports.

Details of how to execute imports are held in ODI's Master Repository and Work Repository.

## **Latency**

Performance is the result of the combination of the hardware and software configuration. The best performance will depend on the specific situation at a retailer.

This deployment method shown has the database schema (containing the MSP database and ODI repositories), XML files, and ODI Standalone Agent on the same server. This reduces latency between the different operations during planogram import and may improve performance.

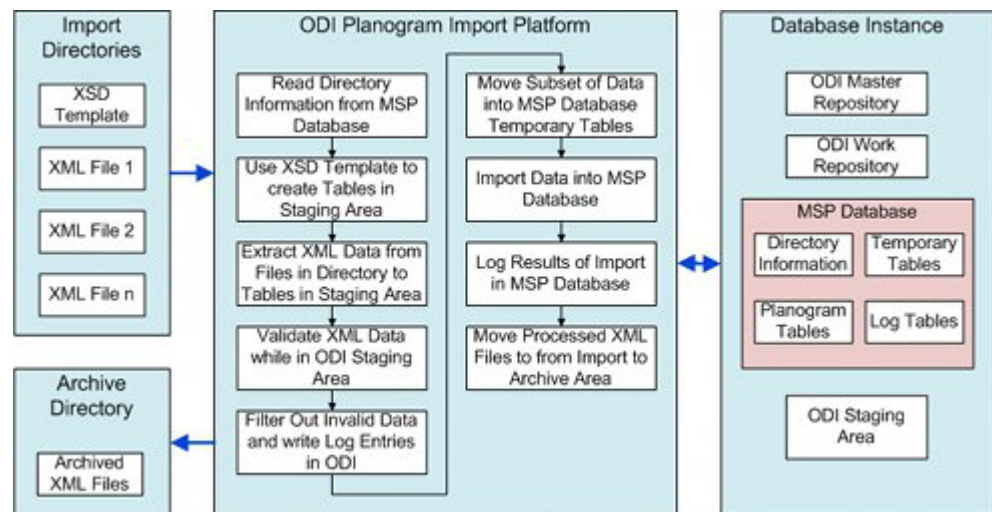
When configuring the ODI planogram solution in this manner, the limitations of the hardware must also be taken into account; factors including the disk I/O, CPU, memory and number of disc controllers may also impact on performance.

Advice on hardware and software configuration may be available from Oracle on request.

## Planogram Import Process Overview

This section gives a high level overview of the ODI Planogram Import process.

**Figure 2-3 Planogram Import Process Overview**



The core of the process is ODI. It functions as follows:

1. A scheduled import process is initiated by the ODI Agent (standalone or local). The agent reads the Master Repository and Work Repository for information on what instructions to execute.
2. Information is read from the MSP database as to which directory contains the required information to import. The data is in the form of one or more XML files containing data on the planograms to be imported.
3. ODI uses the XSD template to create a set of blank tables in the ODI Staging Area in the database instance.
4. Data is extracted from the XML files and placed in the ODI Staging Area.
5. Data in the Staging Area is validated against the XSD to insure that it is in the correct form and contains appropriate values.
6. Any invalid data is flagged as such and is filtered out for import purposes. A series of entries are written to the ODI log tables describing the outcome of each stage in the import process.

7. The data in the staging area is transformed into a form suitable for import into the MSP database. Some data uses a multiple step transformation via temporary tables in the MSP database.
8. Data is imported into the Planogram tables in the MSP database, where necessary referencing 'look-up' data in the temporary tables in the MSP database.
9. The results are logged in the MSP database. This has a list of all changed values (AVTTB\_ODI\_LOG table) and all rejected values (SNP\_CHECK\_TAB table).
10. The processed XML files are moved from the Import to the archive directory.

## Size and Capacity Requirements

Prior to installing the planogram import solution, it is recommended that the hardware requirements be sized. Factors affecting size requirements include:

- The time frame available for planogram import.
- The maximum number of planograms to be imported.
- The quantity of data associated with importing a single planogram.

The amount of data depends on whether just mandatory information is being imported, whether summary (2D) or full (3D) information is being imported, and whether optional information such as financial performance is also to be imported.

It may be necessary to set up a test environment and gain practical experience of planogram import rates prior to specifying the final hardware requirements.

---

---

**Note:** Please contact Oracle Technical Support for further information and assistance on sizing the requirements for Planogram import.

---

---

---

---

## Overview of Planograms

This chapter of the ODI Planogram Import documentation gives basic information on planograms. It is intended for implementers and administrators who are not familiar with the concept.

Those with a background in retail operations can omit this section if required.

### Overview of Planograms

This section gives a brief overview of planogram design and terminology.

### Purpose of Planograms

Planograms (sometimes abbreviated to POGs) are a standardized arrangement of equipment, products and signage designed to maximize sales at a retail outlet. This is an industry standard way of communicating visual merchandising arrangements to 'bricks and mortar' sales outlets.

Planograms are generally designed using specialized software. This software often takes into account a range of information including:

- The psychology of how consumers shop.
- The direction those customers walk past the planogram (left to right/right to left).
- The performance of individual products available for placement within the planogram.
- How best to display and promote the products in the planogram.

Planograms are normally designed at a retail organization's headquarters (or sometimes at the retailer's suppliers). The design takes the form of a diagram showing how the merchandise is to be laid out, together with one or more pages of information giving details of the individual products within the planogram.

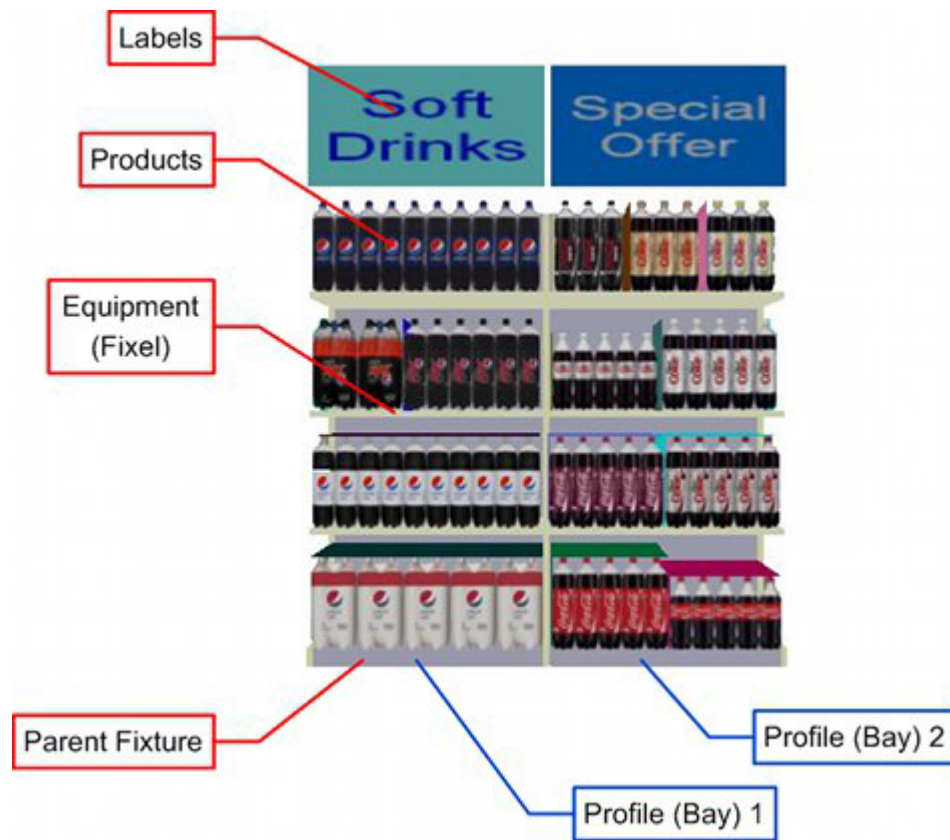
Details of the planogram design are sent to the stores (published) at predetermined times. The stores then implement the planogram by stacking and displaying products on fixtures and shelves in the required format.

Planogram performance is monitored. When necessary the planogram design is revised or replaced, enabling a retail chain to maintain the best possible combination and arrangement of merchandise in its stores.

### Planogram Terminology

Planogram designs have specific terminology associated with them. The diagram below is of a 2 bay planogram. It shows some of the more common terms.

Figure 3–1 Planogram Components



### Planogram

This term refers all aspects of the design including the equipment, products and associated information.

### Parent Fixtures

Planograms are designed to go on fixtures of a specific size and type. Such fixtures are often described as parent fixtures.

### Profile (Bay)

Planograms may be designed to occupy multiple fixtures. Each fixture can be described as a profile or a bay.

### Equipment (Fixels)

Each fixture in the planogram may have items of equipment such as shelves, pegs, dividers or grilles associated with it. Such items of equipment are sometimes described as fixels.

### Products

The planogram contains a number of products, displayed in a carefully designed arrangement.

## Labels

Planograms have signs associated with them. These can be the banner signs above the planogram, shelf edge labelling and other advertising material. This signage is often collectively referred to as labels.

## How Planogram Data is Stored

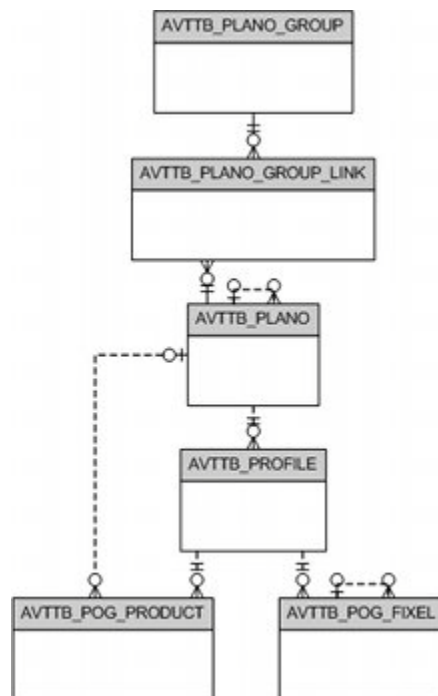
This section specifies how planogram data is stored in the Macro Space Planning Database.

---

**Note:** For a more comprehensive description of the tables and fields involved, please see the Macro Space Planning Data Model.

---

**Figure 3–2 Basic Planogram Tables**



The diagram above shows the most basic tables associated with holding planogram information.

The **AVTTB\_PLANO** table holds header (summary) information on the planogram. Information on specific profiles (bays) is held in the **AVTTB\_PROFILE** table. The **AVTTB\_POG\_FIXEL** table holds information on items of equipment associated with a profile (bay), while the **AVTTB\_POG\_PRODUCT** table holds information on products associated with a specific profile (bay).

The planogram hierarchy is arranged in a series of planogram groups held in the **AVTTB\_PLANO\_GROUP** table. These planogram groups are linked to specific planograms (**AVTTB\_PLANO** table) by the **AVTTB\_PLANO\_GROUP\_LINK** table.

Below is a more detailed discussion of tables used to hold planogram information. For convenience, the tables have been split into the following broad categories:

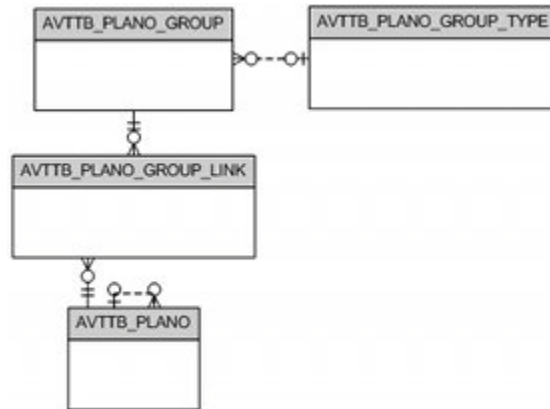
- Planogram Hierarchy

- Planogram Components
- Planogram Information
- Planogram Data

## Planogram Hierarchy

The planogram hierarchy is used to arrange planograms into logical groupings sharing common characteristics.

**Figure 3–3 Planogram Hierarchy Tables**



It uses the following tables:

### **AVTTB\_PLANO\_GROUP**

This table holds a list of planogram groups. These groups are used as containers for planograms sharing a common characteristic. For example the **Wines and Spirits** planogram group might be associated with planograms for **Red Wines, Sparkling Wines, White Wines, Spirits** and **Speciality Liquors**.

### **AVTTB\_PLANO\_GROUP\_LINK**

This is a link table associating specific planograms (**AVTTB\_PLANO** table) with their parent planogram group (**AVTTB\_PLANO\_GROUP** table).

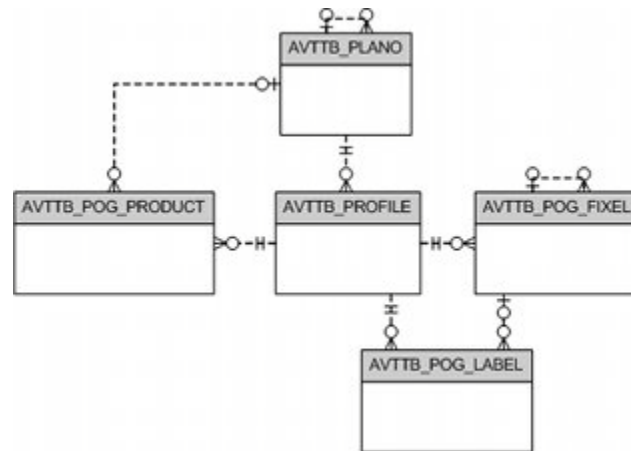
### **AVTTB\_PLANO\_GROUP\_TYPE**

This table is used to associate planograms with specific levels in the Product Hierarchy (**AVTTB\_PRODUCTTYPE** table - not shown). The information is used for planogram publishing purposes.

## Planogram Components

These tables describe the physical components of the planogram.



**Figure 3–4 Planogram Component Tables****AVTTB\_PLANO**

This table holds summary information on the planogram, including the name, size and status.

**AVTTB\_PROFILE**

This table holds information on the size and position of a fixture holding a specific bay in a planogram. There will be a minimum of one profile (bay) per planogram. Some planograms can have multiple bays. This results in one entry for each bay in the planogram within the AVTTB\_PROFILE table.

**AVTTB\_POG\_FIXEL**

This table holds information on the items of equipment associated with a specific profile (bay) within a planogram. It would typically hold information on shelves, pegs and their associated equipment. The number of items of equipment associated with a planogram may range from just a few shelves to dozens of pegs.

**AVTTB\_POG\_PRODUCT**

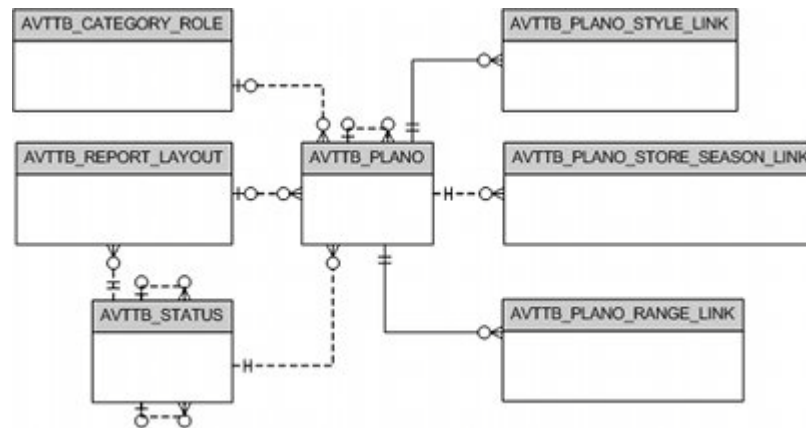
This table holds information on the products associated with a specific profile (bay) within a planogram. Even a simple planogram may contain dozens of products.

**AVTTB\_POG\_LABEL**

This table holds information on any labelling associated with the planogram. Labelling may come in a number of forms including banners above the planogram, shelf edge labelling and promotional material.

**Planogram Information**

These tables hold specific information that associates the planogram with other Macro Space Planning functionality.

**Figure 3-5 Planogram Information Tables****AVTTB\_CATEGORY\_ROLE**

This table holds the category role the planogram is associated with. Category Roles describe the reason the customer might buy a product from the planogram: this might be because it holds staple products, seasonal buys or specialist purchases.

**AVTTB\_PLANO\_RANGE\_LINK**

This link table associates the planogram to a range - most typically a temperature range. (**AVTTB\_RANGE** table - not shown) This information can be used to ascertain whether the planogram is on a compatible type of fixture.

**AVTTB\_PLANO\_STORE\_SEASON\_LINK**

This link table associates a planogram to specific stores or to specific seasons. This information can be used to ensure planograms are customized for specific markets (for example a particular customer demographic) or for a specific time of year.

(Store and season information is held in the **AVTTB\_STORE** and **AVTTB\_SEASON** tables - not shown).

**AVTTB\_PLANO\_STYLE\_LINK**

This link table associates a table with a particular style (**AVTTB\_STYLE** table - not shown). Styles are used within Macro Space Planning as a filter to ensure the planogram is being placed on compatible equipment.

**AVTTB\_STATUS**

The **AVTTB\_STATUS** table holds information on statuses. These are used to define where the planogram is within its life cycle - for example being designed, in service or superseded.

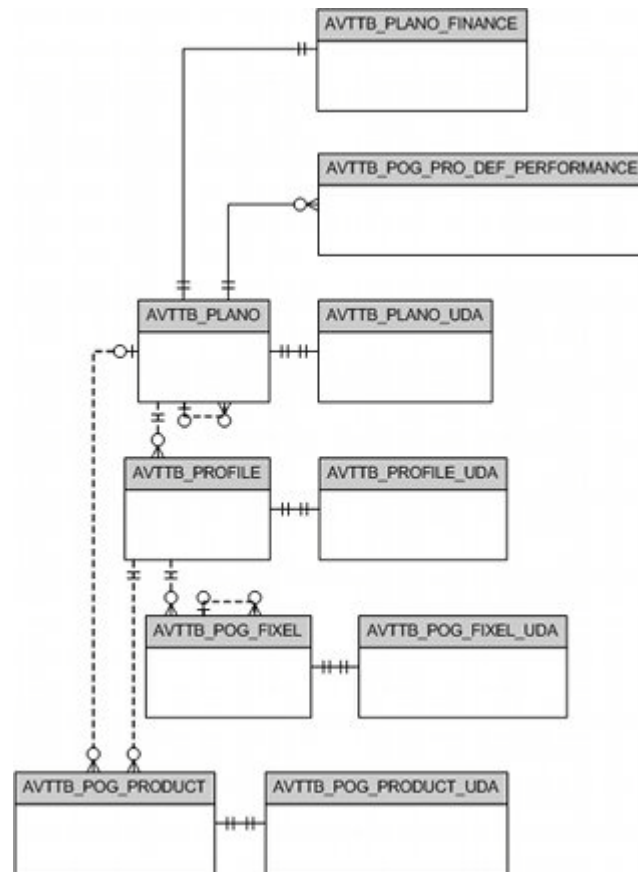
**AVTTB\_REPORT\_LAYOUT**

This table is used to specify the format the information is to be displayed when a planogram design is printed or published. The report layout generally includes both diagrams and tables.

**Planogram Data**

These tables hold information and data concerning the performance of the planogram and its component parts.

Figure 3–6 Planogram Data Tables



The **AVTTB\_PLANO**, **AVTTB\_PROFILE**, **AVTTB\_POG\_FIXEL** and **AVTTB\_POG\_PRODUCT** tables hold information on the components of the planogram. Each of these has data associated with them.

#### **AVTTB\_PLANO\_FINANCE**

This table holds information on the financial performance of the planogram. The time period the data is derived from is customer specific. The information is not store specific.

#### **AVTTB\_POG\_PRO\_DEF\_PERFORMANCE**

This table holds information on the performance of individual products within the planogram. Like the planogram information, the time period is customer specific and not store specific.

#### **AVTTB\_PLANO\_UDA**

This table holds User Defined Attributes for the planogram as a whole. User Defined Attributes are customer specific information that can be configured to meet the internal needs of a retail organization.

#### **AVTTB\_PROFILE\_UDA**

This table holds User Defined Attributes for profiles (bays) within a planogram.

**AVTTB\_POG\_FIXEL\_UDA**

This table holds information on the User Defined Attributes for a specific item of equipment within a planogram.

**AVTTB\_POG\_PRODUCT\_UDA**

This table holds information on the User Defined Attributes for a specific product within a planogram.

---



---

## Overview of Oracle Data Integrator

This chapter provides an overview of Oracle Data Integrator (ODI) for new users of the application. It introduces basic concepts which are useful when installing, configuring and using ODI.

This section describes ODI when installed in a Windows environment. It is also possible to install ODI in a Java (WebLogic) environment. This configuration for ODI is not discussed in this section.

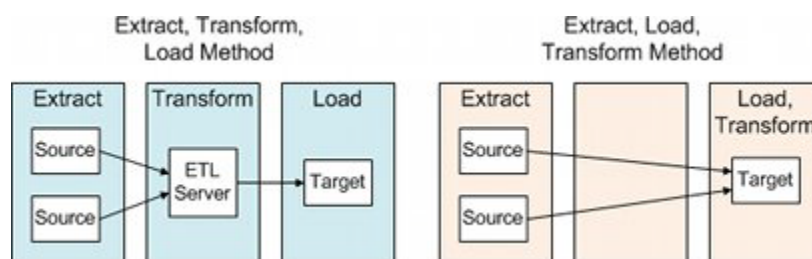
More detailed information is available in the ODI documentation. Two useful documents are:

- *Oracle Fusion Middleware: Getting Started with Oracle Data Integrator*
- *Oracle Fusion Middleware: Developers Guide for Oracle Data Integrator*

### Overview of Data Transfer Methods

ODI is the Oracle corporate solution for transferring data from one location to another, transforming it as required. It uses a different methodology to traditional Extract-Transform-Load (ETL) tools.

**Figure 4–1 Extract, Transform, Load Methodologies**



In a traditional ETL tool, data is extracted from files or databases on the source server and moved to the ETL server. After transformation the data is then exported to the destination server and loaded into the target database.

ODI uses a Extract, Load, Transform methodology where data is moved directly from the source to the destination server. Data is then transformed on this server before being inserted into the target database. This method offers performance advantages over traditional ETL.

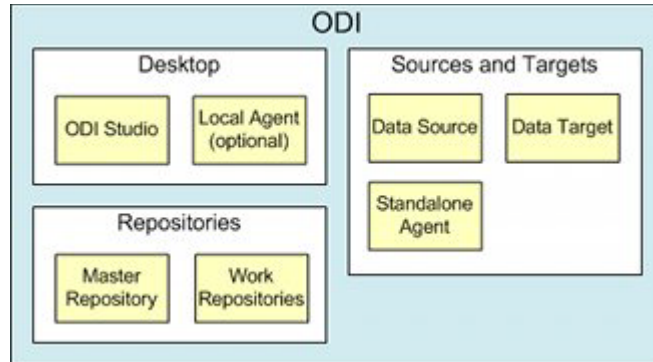
## ODI Architecture

ODI can run on two environments:

- **Windows** - where it typically runs inside a Java Virtual Machine (JVM).
- **Application Server** - where it typically runs in a Java (WebLogic) Environment.

This section discusses ODI when installed in a Windows environment. The architecture is described in the following diagram.

**Figure 4–2 ODI Components**



- **ODI Studio** is where the details of imports are configured. ODI Studio is normally installed on a PC and can be paired with a local agent. This local agent is used to execute imports at scheduled times.
- **Repositories** are where ODI stores the information on how to configure data imports. Once stored, this information can be acted on by an agent without further input from ODI Studio.
- **Sources and Targets** are where data is taken from and moved to. Scheduled processes are initiated by an agent (local or standalone) This may be done using scheduling information that was previously configured in ODI Studio, or via an external scheduling tool.

The overall process may be considered as follows:

1. The method to carry out a data import is configured in ODI Studio.
2. Once configured, the method is stored in the Master and Work Repositories.
3. The import is executed at scheduled times by an Agent that references the Master and Work Repositories for details of the method.

Each component is discussed in more detail below.

### ODI Studio (Desktop Option)

ODI Studio is used for design, administration, and monitoring:

- **Design:** ODI Studio is used to configure the details of the data imports. The methods are provided as compiled scenarios that can be executed by retailers.
- **Administration:** ODI Studio can be used at retailers to determine the schedule which the data imports will be executed, together with the environments those imports should be executed in and the level detail used for logging results.

- **Monitoring:** The Operator Navigator can be used at retailers to review the logged results of each data import.

ODI Studio can be used to manually initiate a data transfer. However, it is more normal to initiate data transfers to a schedule using a local or standalone agent. If local or standalone agents are used, it is not necessary for ODI Studio to be open for the data transfer to occur.

## Navigators

ODI Studio consists of four modules (or Navigators). Each has a different purpose.

### Security Navigator

Security Navigator is used to configure access permissions for data accessed and manipulated by ODI.

---

---

**Note:** This documentation uses the default ODI Login and Password for convenience. It is recommended that the Security Navigator be used to change these when the ODI Planogram Import solution is installed at a retail organization.

---

---

### Topology Navigator

Topology Navigator is used to specify details of the physical and logical architecture accessed by ODI. In the context of the ODI Planogram Import solution it is used to specify the import source and target database for that information.

### Designer Navigator

Designer Navigator is initially used to configure the detail of how each item of data is extracted, loaded, and transformed. It can then be used to schedule and execute data transfers.

---

---

**Note:** The Designer Navigator can only connect to Master or 'Development' type Work Repositories.

---

---

### Operator Navigator

Operator Navigator is used to specify the details of the operations to be executed at run time. For the ODI Planogram Import solution, it can be used to view the ODI logs to determine how the imports have progressed.

### Local Agent

The local agent can be installed in parallel with ODI Studio. It is a tool for executing the instructions required to carry out the data transfer. It can be used to execute imports at specified times.

## Repositories

The repositories are the core of the system. They are schemas within a database instance where information on how to execute the data transfer is stored. This data includes security information, locations of servers and other sources of data and how to transform that data. Repositories are of two types:

## Master Repository

The Master Repository holds high level information on how the solution has been configured. This includes:

- Security information on who can access what via ODI.
- Topology information including available technologies, servers, schemas, etc.
- Versioned and archived information.

Many implementations only use a single active Master Repository, although it is possible to implement the ODI Planogram Import solution using multiple Master Repositories.

## Work Repositories

Work Repositories contain the details of Extract, Load and Transform (ELT) projects. This information can include:

- Models describing the source data that is to be transformed and the schema it is to be transferred to. This information includes descriptions of tables and columns, together with information on other database objects such as constraints.
- Projects containing the business rules for transforming the data.
- Scenarios. These consist of compiled information on how and when to execute an ELT instance.

Work Repositories are of two types:

### Development Repositories

Development Work Repositories contain the full range of information to develop and configure an ELT project. This includes models, projects, and scenarios.

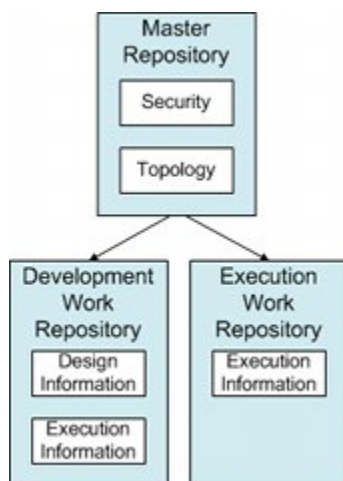
### Execution Repositories

Execution Work repositories only contain the scenario information used at run time. Execution Repositories are normally used in production.

## Relationship Between Repositories

The relationship between the varying repository types can be seen in the following diagram:



**Figure 4–3 Repository Relationships**

- The Master Repository holds overall information on the data transfer.
- The Development Work Repository enables information on the details of the data transfer to be both configured and executed.
- The Execution Work Repository only holds the information required to carry out the data transfer.

---



---

**Note:** If a user connects to an Execution type Work Repository, the Designer Navigator is not be enabled for use.

---



---

## Agents

The Agent is a tool for executing the instructions required to carry out the data transfer. In the Windows environment, it runs inside a Java Virtual Machine (JVM). Agents can be deployed in one of two ways:

### Local Agent

Here, the agent is created on the same machine as ODI Studio. When ODI Studio executes a scheduled task, it can use this agent. It is not necessary for ODI Studio to be running for the local agent to execute a data transfer.

One use for a local agent is when the ODI Planogram Import process is being tested - it may be convenient to have a local agent when running scheduled imports.

### Standalone Agent

The standalone agent is generally created on an separate computer (or server) to ODI Studio. When activated, this agent is capable of executing scheduled tasks. It then connects to the necessary servers and executes the data transfer. Finally it writes results and log information back to the Work Repository.

A common use for a standalone agent is to run scheduled production planogram imports.

### Methods of Executing Data Transfers using Agents

A data transfer initiated by a local or standalone agent works as follows:

1. The agent activates the task at a scheduled time.
2. The agent connects to the Work Repository and retrieves the necessary instructions.
3. It then connects to the necessary servers and executes the data transfer.
4. Finally, the agent writes results and log information back to the Work Repository.

---

---

**Note:** Data transfers can also be initiated manually from ODI Studio.

---

---

### **Scheduling Agent**

There are two basic ways of scheduling agents to execute a transfer:

#### **Using ODI's Scheduling Tool**

Using this method, a schedule is specified using ODI Studio's scheduling tool. The agent is left running continuously. Each time the agent reaches a time specified in the scheduler, it will execute a job or scenario.

#### **Using an External Scheduling Tool**

An alternative method is to use an external scheduling tool. ODI Studio is configured to activate a data transfer each time an agent activates. The agent is then turned on or off by an external scheduling tool using command line prompts or calling batch/shell scripts.

---

## Overview of Installing the Target Bridge

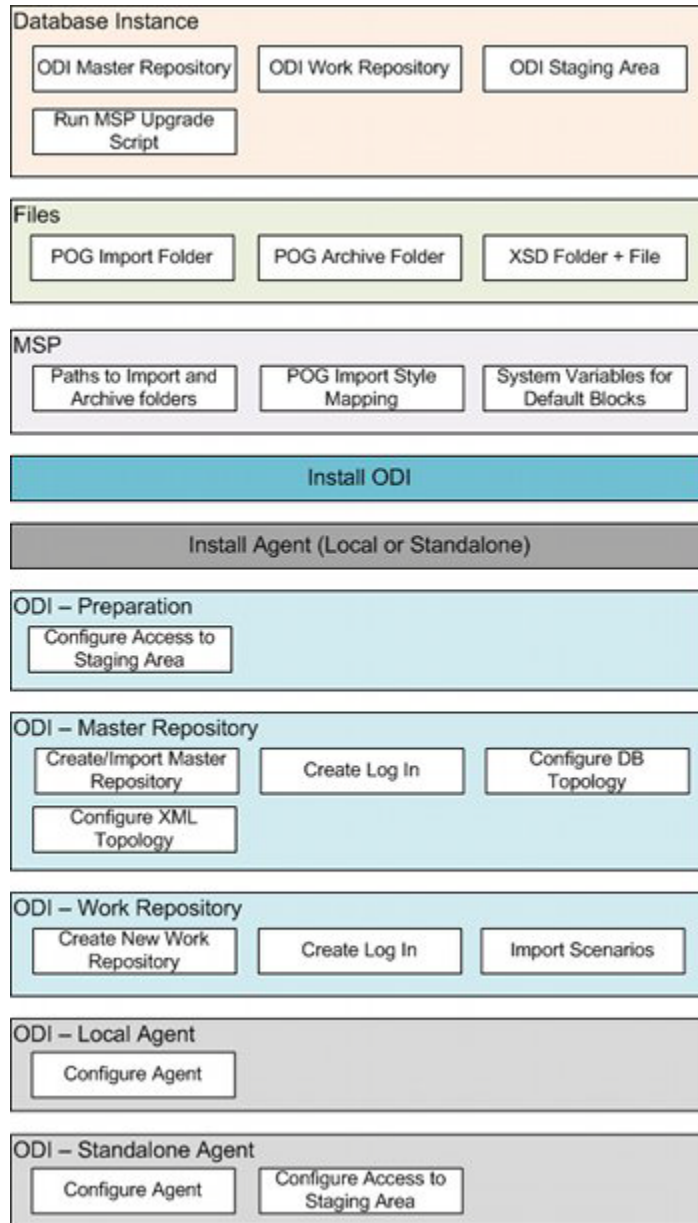
This gives an overview of how to install and configure the Target Bridge part of the Planogram Import solution using ODI Studio in a desktop (Windows) environment. It involves the following stages:

- Making the required changes to the database instance
- Creating the required folders on the file server
- Configuring settings within Macro Space Planning
- Installing ODI Studio
- Installing the necessary Local or Standalone Agents
- Configuring ODI Studio
- Configuring the Local Agent
- Configuring the Standalone Agent

Tasks that can be accomplished at the same time have been grouped together

After reading this overview, please consult the following chapters for details of how to execute each stage.

**Figure 5–1 Implementation Flowchart**



## Database Instance

The following actions need to be carried out in a database instance.

### Create Schema for ODI Master Repository

This schema holds information for ODI's Master Repository. It is used to hold high level information on objects ODI can connect to. The schema is blank at this stage and can be populated with information when ODI Studio is configured later.

### Create Schema for ODI Work Repository

This schema holds information for ODI's Work Repository. It is used to hold information on how ODI extract, loads and transforms information. The schema is

blank at this stage and will only be populated with information when ODI Studio is configured later.

### **Create Schema for ODI Staging Area**

This schema holds temporary tables created and deleted during the course of a data import. The schema is blank and will only be populated during the course of a data import.

### **Run Upgrade Script against MSP Database**

This upgrade script adds additional tables, stored procedures, and functions used for the data import process to the MSP database.

## **Files**

The following actions need to be carried out on a server. One method to reduce latency to the lowest possible level for production imports is to create these folders on the same server holding the database instance. When doing so consideration must be given to the constraints imposed by the server hardware.

### **Create an import folder for the XML Files holding data to be processed**

This folder serves as a container for the XML files to be processed. The XML files are created directly here by the Source Bridge, or are copied here after any Source Bridge validation operations.

### **Create a folder for the archived XML Files**

After the data from the XML files has been processed, files will be moved irrespective of Success/Failure to avoid situation of processing of the same failed XML file repeatedly when POG importer is executed with a valid polling interval.

### **Create a folder to hold the XSD file used to create the tables in the ODI Staging Area**

The XSD file is used to create a set of temporary tables in the ODI Staging area. These tables are then populated with data during the import process and deleted at the end of it. To do this, a folder is created and a copy of the pertinent XSD file placed in it.

---

**Note:** The Import and archive folders need to be installed locally to the entity initiating the data transfer. These same local folders can be used if scheduled imports are to be executed using a local agent.

For test imports initiated manually from ODI Studio installed on a desktop, the folders need to be on that desktop.

For production imports initiated by a standalone agent on the database server, the folders need to be local to that standalone agent.

---

## **Macro Space Planning**

The following actions need to be carried out in the Administration Module of MSP.

### **Ensure Base Data is Correct**

Certain data will probably already be in the MSP database. This should be checked to verify there it maps correctly with the data to be imported from the planogram design

software. In some cases it may be necessary to verify that the correct defaults are set or look up values have been assigned. This base data is:

**Administration Module**

- Category Roles
- Hatch Styles
- Manufacturers
- Ranges
- Seasons
- Statuses
- Strategies
- Tax Codes
- Units
- User Defined Attributes

**Fixture Studio Module**

- Fixture Groups (Block Groups)
- Product Shapes (Product Blocks)
- Sizes

**Product Studio**

- Display Style Types
- Materials
- Product Hierarchy

---

---

**Note:** See the Macro Space Management *Administration Module*, *Fixture Studio* and *Product Studio User Guides* for more information.

---

---

## Configure Planogram Import Specific Information

Some information will require specific configuration of ODI Planogram Import. This information is:

**Set Planogram Import Style Mappings**

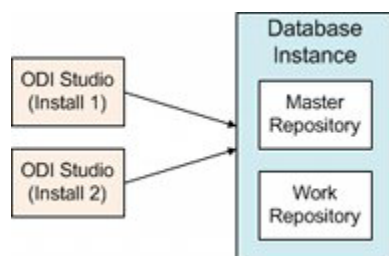
The Planogram Import Style Mappings are used as 'look-ups' to associate information specific to MSP with data in the process of being imported.

**Set System Variables**

Specific MSP system variables are referenced by ODI to identify the path to the import and archive directories, together with the default fixture, shelf and product blocks.

## Install ODI Studio

ODI Studio is generally installed on a desktop (although it can be installed on a server). It is possible to install ODI Studio on multiple machines, each install of ODI Studio pointing to the same set of Master and Work Repositories.

**Figure 5–2 Multiple Installs of ODI Studio**

This situation might occur if the first install of ODI studio was used to set up and configure the ODI Planogram Import solution and the second install was used by another person on a different machine to schedule, run and verify planogram imports.

## Install Agents

ODI Studio is capable of manually invoking a data transfer, but running imports to a predetermined schedule requires an agent. This agent may be installed in one of two forms:

- **On the same machine as ODI Studio**

An agent installed in this manner is referred to as a local agent.

- **On the Database Server**

An agent installed in this manner is referred to as a standalone agent.

Either of these agents can be used to initiate the import to a schedule, but the folders holding the XML files for import and archiving must be local to the agent initiating the import.

## Configure ODI

After any local and standalone agents have been installed, the next stage is to configure ODI.

### Create Access to the ODI Staging Area

The ODI Staging Area is used to temporarily hold information from the XML files while it is validated against the XSD file.

#### Create Access to Staging Area

Access to the Staging Area for ODI Studio or ODI Console is created by putting the **ExtXML.properties** file in a directory accessed by ODI. Data from this file will enable ODI to connect to the database instance holding the Staging Area schema.

### Create/Import the ODI Master Repository

The Master Repository holds information on objects ODI can connect to. Once ODI is installed, it can be created and configured. The required stages are:

### **Create/Import the Master Repository**

The Master Repository can be created via the standard ODI tool. During the creation process a Zip file (supplied by Oracle) is imported containing configuration information for the master registry.

### **Create Login for the Master Repository**

The login that is created allows users to connect to the Master Repository during normal ODI operations.

### **Configure Target Database Topology**

This action specifies the details for the database ODI is to import data into.

### **Configure XML Topology**

This action specifies the location and details of the XSD file used to validate the XML data during import.

## **Create the ODI Work Repository**

The Work Repository holds information on how data is to be extracted, loaded, and transformed.

### **Create the Work Repository**

The Work Repository can be created using the standard ODI functionality. It is generally configured as an 'Execution' type work repository. This is because the planogram import method is delivered as compiled scenarios that cannot be modified by a customer.

### **Create Login for the Work Repository**

The login that is created allows users to connect to the Work Repository during normal ODI operations.

### **Import Scenarios**

The scenarios contain detailed instructions as to how data is to be extracted from the XML files supplied by the source bridge, loaded to the target database and transformed. The scenarios are supplied by Oracle.

## **Configure the Agents**

The agents can be installed on the same machine as ODI Studio (local) or on a server - for example the database server (standalone).

After the agents have been installed, they require configuration. They can then be used to initiate and execute scheduled planogram imports.

Configuring the agents requires adding parameters to the odiparams.bat batch file. Standalone agents also require access configured to the ODI Staging Area.



---

---

## Configuring the Database Instance

This chapter describes how to configure the database instance in preparation for installing and configuring the rest of the solution. In order to do this, two levels of permissions are required:

- System DBA privileges for the database instances to create the users and roles for the schemas required for the Master Repository, Work Repository and Staging Area.
- User privileges to run an upgrade script against the Macro Space Planning (MSP) database.

### Configuring the Database Instance

This section covers the process of configuring a database instance.

#### Creating the Schemas required by ODI

It requires three database schemas be created. These are:

- **Schema to hold the ODI Master Repository**  
This Repository will hold high level information how the planogram import solution is configured.
- **Schema to hold the ODI Work Repository**  
This Repository holds detailed information on how data is to be processed during the planogram import.
- **Schema to hold the ODI Staging Area**  
This schema holds planogram import data being validated after extraction from the source XML file and before it is inserted into the Macro Space Planning database.

The person accomplishing this task must log in as SYS with SYSDBA privileges (or an account with similar privileges) in the database instance that these schemas are created in. If they do not have these privileges, they may not be able to grant all the roles to the users they create. This is done as follows:

1. Open SQL Plus.
2. When asked for a user name enter `sys@<Database Name>` as `sysdba`
3. Enter the password for the sys user.

**Figure 6–1 SQL Plus - Logging in as Sysdba**

```

SQL*Plus: Release 11.2.0.1.0 Production on Wed Apr 11 09:30:09 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: sys@SPACEPLAN as sysdba
Enter password:
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> _

```

In the above example, the user has logged in as the **sys** user to the **SPACEPLAN** database instance with **sysdba** privileges.

### Creating the Schema for the ODI Master Repository

Open SQL Plus and log in. The users and their associated privileges can be created using the following syntax:

Create user [user name] identified by [password] temporary tablespace [name of tablespace] default tablespace [name of table space];

Grant connect, resource to [username];

Grant execute on dbms\_lock to [username];

#### Example

An example of creating the **Master Repository** is as follows:

Create user **POG\_IMPORT\_MASTER** identified by **Password\_Master** temporary tablespace **TEMP** default tablespace **USERS**;

Grant **connect, resource** to **POG\_IMPORT\_MASTER**;

Grant **execute on dbms\_lock** to **POG\_IMPORT\_MASTER**;

### Creating the Schema for the ODI Work Repository

The users and their associated privileges can be created using the following syntax in SQL Plus.

Create user [user name] identified by [password] temporary tablespace [name of tablespace] default tablespace [name of table space];

Grant connect, resource to [username];

Grant execute on dbms\_lock to [username];

#### Example

An example of creating the **Work Repository** is as follows:

Create user **POG\_IMPORT\_WORK** identified by **Password\_Work** temporary tablespace **TEMP** default tablespace **USERS**;

Grant **connect, resource** to **POG\_IMPORT\_WORK**;

Grant **execute on dbms\_lock** to **POG\_IMPORT\_WORK**;

## Creating the Schema for the ODI Staging Area

The users and their associated privileges can be created using the following syntax in SQL Plus.

```
Create user [user name] identified by [password] temporary tablespace [name of tablespace] default tablespace [name of table space];
```

```
Grant connect, resource to [username];
```

```
Grant execute on dbms_lock to [username];
```

### Example

An example of creating the **Staging Area** is as follows:

```
Create user XMLHOLD identified by Password_XMLHOLD temporary tablespace TEMP default tablespace USERS;
```

```
Grant connect, resource to XMLHOLD;
```

```
Grant execute on dbms_lock to XMLHOLD;
```

---



---

**Note:** The schema name for the ODI Staging Area must be exactly seven characters long. This is because the maximum limit for Oracle table names is 30 characters. Oracle dynamically truncates the table name if it exceeds this value. Oracle databases automatically add the schema name as a prefix to the table name. Having a schema name of fixed size allows the planogram import functionality to cater for this automatic truncation.

---



---



---



---

**WARNING:** If the schema name of the ODI stage area is not exactly seven characters long, planogram import could fail.

---



---

## Running the Upgrade Script Against the MSP Database

The upgrade script adds some additional tables, modifies some existing tables and adds a number of new stored procedures and functions required for planogram import.

The following tables are added:

- **AVTTB\_ODI\_IMPORT\_MAP:** this table is referenced by ODI during the import process.
- **AVTTB\_ODI\_LOG:** this table holds the results of all attempts to insert data into the database.
- **PLANOGRAM\_TEMP:** this table is used to temporarily hold planogram data during planogram import.
- **PLANOGRAMGROUP\_TEMP:** this table is used to temporarily hold information on planogram groups (the planogram hierarchy) during planogram import.

---



---

**Note:** ODI adds a table called SNP\_CHECK\_TAB after the first planogram import. It is not added by the script.

---



---

### **Prerequisites**

The person running the script must have the appropriate privileges for the MSP database schema within the database instance.

### **Running the Upgrade Script**

The 13.2.5.00 POG Importer DB Script GA.sql upgrade script is available with the package. It can be found in the DB Script folder.

In order to run this upgrade script, log into the database as the MSP database user. The script can then be run using any standard query and management tool (such as Oracle's SQL Developer). After executing the script, verify that it has executed without errors.

---

---

## Creating the Data Folders

This section of the Oracle Data Integrator (ODI) Planogram Import solution documentation describes how to configure the folders required to hold data for the planogram import. Installing these folders requires the appropriate permissions on the computer they are to be installed on.

### Creating the Required Data Folders

This section specifies how to create the required folders. The paths to the folders for the imported and archived XML files are specified in the Macro Space Management (MSM) system variables.

### Creating the folders for Import and Archived XML files

Two folders are required to hold XML files pertinent to planogram import.

#### Import Files

This folder is required to hold the XML files that are used as the data source for the import.

#### Archive Files

This folder holds the files that have been processed during data import.

### Creating the Folders

These files must be created local to the ODI functionality used to execute the ODI import.

- If the Planogram import is executed from ODI Studio, the folders for the import and archive files must be local to the machine ODI Studio is installed on.
- If the Planogram import is executed from ODI Standalone Agent, the folders for the import and archive files must be local to the machine the Standalone Agent is installed on.

---

---

**Note:** Care must be exercised with these file locations. If the ODI Planogram Import solution is validated using ODI Studio for test purposes, the files must be local to that machine. If it is decided to use and ODI Standalone Agent to bring in 'live' planogram imports, a new pair of folders must be created and the MSM System Variables changed accordingly.

---

---

It is therefore suggested that folders for the import and archive directories for the XML files holding planogram be created on each machine that is used to invoke a planogram import. The MSM system variables can then be updated as required so they reflect the machine being used.

See the section on Configuring the MSM Administration module for details on how to configure the system variables.

Note the name and path of the folders, together with the machines they have been created on.

## Creating the Folder for and Placing the XSD File

The XSD file is used to create the required tables in the ODI Staging Area prior to each planogram import. The folder must be created on a server that will be accessible to the ODI application. Access information will be added to ODI Studio when configuring that application.

Create a folder to hold the XSD and note the path and name.

The XSD file is supplied with the ODI Planogram Import functionality. It is in the folder called **Data Model**. This contains a file called **XMLTemplate.xsd**. Place a copy of this file in the folder that has just been created.

---

# Configuring the MSM Administration Module

This chapter describes how to make the necessary changes in the Macro space Management (MSM) Administration Module.

The parameters entered are referenced by ODI during the planogram import process. This data is:

## Configuring Planogram Import Styles

Macro Space Planning uses Styles to verify that it is permissible to place one item of equipment on another or products on specific types of equipment. This functionality does not generally exist in the planogram design software, so the ability to assign this information has to be configured prior to executing planogram imports.

## System Variables specifying Folder Location for Import and Archived XML Files

ODI references this information to determine where the XML files holding information on the planograms to be imported are located. After import, ODI moves the processed XML files to the archive directory.

## System Variables Specifying some of the Default Variables for the Import Process

These system variables define a number of parameters referenced during the import process.

## System Variables Defining the Default Blocks

These default blocks are used for fixtures, shelves and products if ODI fails to find any other default blocks specified in the Planogram Import Styles dialog box.

## Installation Permissions

Prerequisites for making changes in the MSM Administration module are:

- The supplied upgrade script must be run against the Macro Space Planning (MSP) database. See [Running the Upgrade Script Against the MSP Database](#).
- The person making the changes must have access rights to the Administration module.

## Ensure Base Data is Correct

Certain data will probably already be in the MSP database. This should be checked to verify there it maps correctly with the data to be imported from the planogram design

software. In some cases it may be necessary to verify that the correct defaults are set or look up values have been assigned. This base data is as per the list below:

## Administration Module

### **Category Roles**

The list of Category Roles can be configured by selecting the option from the Merchandising Menu. These category roles should match those used in Category Management.

### **Hatch Styles**

Hatch Styles can be configured by selecting the option from the Planning menu.

### **Manufacturers**

Manufacturers can be configured by selecting the option from the General menu.

### **Ranges**

Ranges can be configured by selecting the option on the Merchandising menu.

### **Seasons**

Seasons can be configured by selecting the option on the Merchandising menu.

### **Statuses**

Statuses can be configured by selecting the option on the General Menu

### **Strategies**

Strategies can be configured by selecting the option on the Merchandising menu. These strategies should match those used in Category Management.

### **Tax Codes**

Tax codes can be configured by selecting the option on the General menu.

### **Units**

Units can be configured by selecting the option on the General menu.

### **User Defined Attributes**

User Defined Attributes can be configured by selecting the option on the General menu.

## Fixture Studio

### **Fixture Groups (Block Groups)**

Fixture Groups can be configured using the Group icons on the toolbar.

### **Product Shapes (Product Blocks)**

Product blocks should be created by someone with autoCAD expertise then can be imported into Fixture Studio. After import, they can be assigned to products in the Size and Shape tab of the Display Styles dialog box in Product Studio.



**Sizes**

Sizes are configured using the Sizes option on the Edit menu. Once configured, they are assigned to equipment in the Size tab of the Block Details dialog box.

**Product Studio****Display Style Types**

Display Style Types can be configured using the option on the View menu.

**Materials**

Materials can be configured using the option on the View menu.

**Product Hierarchy**

The product hierarchy is generally imported. The form it has being imported in must match the form of the product hierarchy associated with the planograms being imported.

**General Information**

When configuring the base information, the following applies:

- As well as directly configuring information, it is also possible to import the required information. An example would be importing lists of manufacturers via the Data Importer module.
- More information is available in the Macro Space Management *Administration Module, Fixture Studio* and *Product Studio User Guides*.

**Configuring ODI Planogram Import Specific Information**

The following information requires specific configuration for planogram import.

**Configuring Planogram Import Styles****Purpose**

Macro Space Planning uses styles to validate that it is permissible to place one item of equipment onto another. The Planogram Import Styles dialog box is used to assign MSP styles to the classes of equipment being imported from the third party planogram database.

During planogram import, the XML file generated by the source bridge will contain a code used to describe a general class of equipment used in the third party planogram database. During import, a MSP style will be assigned to this third party code, adding an item of data required by MSP functionality.

**Preparation**

Prior to entering the information in the Planogram Import Styles dialog box, implementers may find it useful to create a preliminary table identifying:

- The general name of a class of equipment in the third party software design application. Examples include Open shelves, pegs, etc.
- A code or primary key identifying the class of equipment in the third party planogram design software.

- A default block to use in case a precise match for an item of equipment is not available in the MSP database.
- The style assigned to the equivalent class of equipment in the MSP database. (Styles are configured using the Styles option from the Merchandising menu in the Administration Module).

**Table 8-1 Example Block Mapping Table**

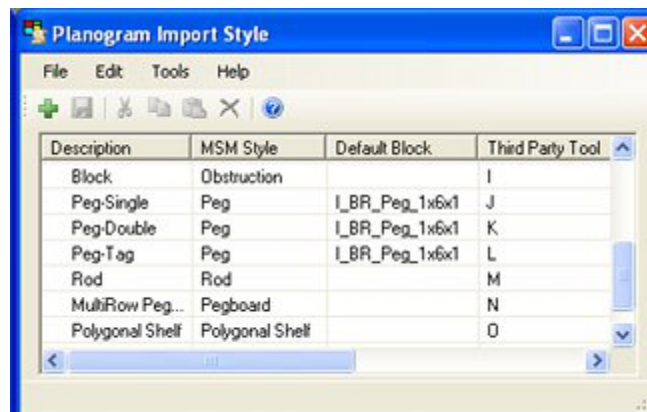
Third Party Equipment Class	Third Party Equipment Code	MSP Default Block	MSP Style to be assigned
Open Shelf	ABC123	I_Stretchable_Shelf	Shelf
Single Peg	XYZ789	I_Stretchable_Peg	Peg

**Note:** It is recommended that default blocks be made stretchable. For more information on making stretchable blocks refer to the *Fixture Studio User Guide*.

### Implementation

The Planogram Import Styles dialog box is accessed via the **Merchandising** menu in the MSM Administration module.

**Figure 8-1 Planogram Import Style dialog box**



To configure Planogram Import Styles:

1. Set the name of third party tool in the Options dialog box accessed from the **Tools** menu in the Planogram Import Style dialog box. This information appears as a column header.

**Figure 8–2 Options dialog Box - Planogram import Styles**

If more than one planogram design application is in use, multiple import sources can be specified.

---

**Note:** If the planogram design application is not in the predefined list, the list can be configured directly in the AVTTB\_POG\_SOURCE table.

---

2. Enter the following details in the Planogram Import styles dialog box:
  - The general name of a class of equipment in the third party software design application.
  - The style assigned to the equivalent class of equipment in the MSP database. (Styles are configured using the Styles option from the **Merchandising** menu in the Administration module).
  - A default block to use in case a precise match for an item of equipment is not available in the MSP database. This column in the dialog box contains a drop down list with all items of equipment assigned that style in the MSP application - the appropriate item of equipment can be selected from it.  
The item of equipment is converted to a BLK\_ID when stored in the database.
  - A code or primary key identifying the class of equipment in the third party planogram design software.
3. After all data has been entered, click **Save** in the Planogram Styles dialog box. Close the dialog box.

---

**Note:** During import, if a default block is not found in the Planogram Import Style dialog box, the block defined in the DEFAULT\_IMPORT\_FIXEL\_BLK\_ID system variable will be used for fixtures and fittings; the DEFAULT\_IMPORT\_FIXEL\_BLK\_ID for shelves, etc.

---

Information in the Planogram Import Styles dialog box is stored in the AVTTB\_STYLE\_IMPORT\_SOURCE\_LINK and AVTTB\_STYLE\_IMPORT\_MAP tables.

### Stretchable Blocks

Because styles are mapped to items of equipment, and those items of equipment may be multiple sizes, it is recommended that the default block selected be stretchable. This

enables it to be 'stretched' to the correct size when the planogram using that item of equipment is placed. For more information, see the *Macro Space Management Fixture Studio User Guide*.

## Configuring the System Variables

This section describes how to configure the system variables associated with planogram import.

### Purpose

These system variables described below are referenced by ODI during planogram import to identify directory locations and default values.

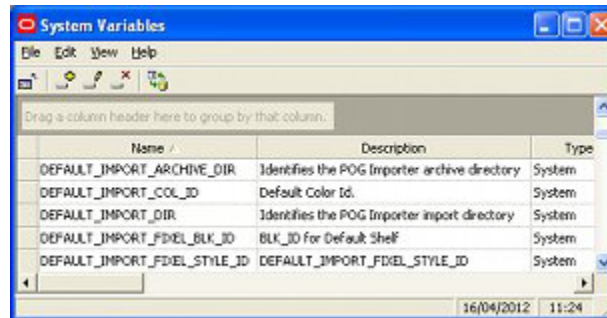
### Prerequisites

Users must have access rights to the MSM Administration module.

### Accessing and Using the System Variable Dialog Box

The System Variable dialog box is accessed from the **General** Menu in the MSM Administration module.

**Figure 8–3** MSM Administration Module - System Variable Dialog Box



Data is entered by selecting the correct system variable then entering the data in the Value column and pressing **Return**.

---

**Note:** Pressing **Return** after entering data is required to write the data back to the database.

---

### Configuring System Variables for the Import and Archive Directories

Two directories are required to store the XML files that hold the data for the planogram import: an Import Directory to hold the XML files prior to processing and an Archive Directory to hold the XML files after successful processing. The paths to these directories are held in MSP system variables. These are:

- **DEFAULT\_IMPORT\_DIR:** this directory holds the XML files prior to processing.
- **DEFAULT\_IMPORT\_ARCHIVE\_DIR:** XML files are automatically transferred to this directory after successful processing.

In this example, the path to the Import folder is **D:\MSP\_Data\POG\_Import\** and the path to the Archive folder is **D:\MSP\_Data\POG\_Archive\**.

---



---

**Note:** These paths terminate with a back slash. It is recommended this be included to avoid potential problems with the functionality.

---



---

The ODI Planogram Import references these system variables during the planogram import process to identify where to find the files containing import data and when to move the files after successful processing.

### Configuring System Variables for the Planogram Import Defaults

Three system variables hold values for parameters that will be assigned as the default values should this information not be contained in the planogram import data. These System Variables are:

- **POG\_IMPORT\_FILTER\_OPTION:** this system variable specifies whether data is to be validated prior to planogram import and whether log files are to be written.
- **DEFAULT\_IMPORT\_MAT\_ID:** this system variable specifies the default material to be assigned to items of equipment. It is used if no material is imported with the planogram definition.

Materials are configured in the MSM Product Studio Module (View menu) and assigned to equipment in Fixture Studio (Category Tab - Block Details dialog box). Material information is held in the **AVTTB\_MATERIAL** table - the value of the **DEFAULT\_IMPORT\_MAT\_ID** system variable is that of the **MAT\_ID** field.

- **DEFAULT\_IMPORT\_COL\_ID:** this system variable specifies the default color to be assigned to products at display style level. It will be used if no product color is imported with the planogram definition.

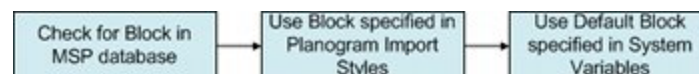
Colors are assigned to products in the **Size and Shape** tab of the Display Style dialog box in Product Studio. Color information is held in the **AVTTB\_COLOUR** Table - the value of the **DEFAULT\_IMPORT\_COL\_ID** system variable is that of the **COL\_ID** field.

### Configuring System Variables for the Default Blocks

#### Purpose

The logic for selecting a block for the planogram operates as follows:

**Figure 8–4 Logic for Default Blocks**



1. The ODI Planogram Import process checks to see if there is an exact match between an item for the planogram as specified in the third party planogram software and a specific block held in the MSP database. If such a block exists, it is used.

For example, if the parent fixture for the planogram being imported is **Example\_Block\_72x24x36** and a similar block exists in the MSP database, this block will be associated with the imported planogram.

2. If there is not an exact match, the ODI planogram Import process will next look to see if a default block has been specified in the Planogram Import Style dialog box.

If a default block has been specified, this block is associated with the imported planogram.

---

**Note:** The Planogram Import Style dialog box only holds default blocks for Fixtures and Fittings.

---

3. If there is no match for the item associated with the planogram being imported, the block default to that specified in one of the three system variables:
  - **DEFAULT\_IMPORT\_FIXTURE\_BLK\_ID:** this system variable holds the default block to be used if no match can be found for the parent fixture for the planogram during planogram import and no default fixture has been configured in the Planogram Import Styles dialog box in the Administration Module.
  - **DEFAULT\_IMPORT\_FIXEL\_BLK\_ID:** this system variable holds the default block to be used if no match can be found for an item of planogram equipment during the import and no default shelf object has been configured in the Planogram Import Styles dialog box in the Administration module.
  - **DEFAULT\_IMPORT\_PRODUCT\_BLOCK\_ID:** this system variable holds the default product block to be used to represent planogram products at Display Style level in the Merchandiser module.

**Prerequisites**

In order to configure these system variable, users will also need access to the Fixture Studio module to identify the block IDs (BLK\_IDs) of the blocks to use as defaults.

**Preparation**

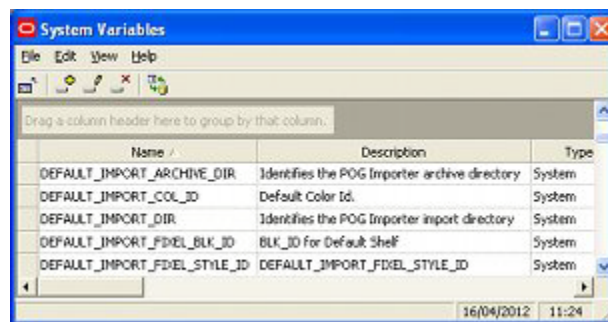
In order to enter the required system variable values, users must know:

- The Block IDs (BLK\_IDs) for the default blocks.
- The paths to and names of the directories.

**Implementing**

To configure these system variables, open the System Variables dialog box from the General Menu of the Administration module.

**Figure 8–5 MSM Administration Module - System Variable Dialog Box**



To enter the value for the system variable:

1. Navigate to the row containing the pertinent system variable.

2. Enter the required value in the **Type** field. Press **Return** to confirm.

---

---

**Note:** Pressing Return is required to save the value back to the database.

---

---

3. When all required items have been entered, exit the dialog box.

**Stretchable Blocks** Because the default blocks could be used for multiple items of equipment, and those items of equipment will be different sizes, it is recommended that the default block selected be stretchable. This will enable it to be 'stretched' to the correct size when the planogram using that item of equipment is placed. For more information see the *Macro Space Management Fixture Studio User Guide*.





---

---

## Installing ODI Studio

This chapter gives basic instructions on how to install an instance of ODI Studio. In order to install this application, you must first download the latest version from the Oracle Technology Network: <http://www.oracle.com/technetwork/index.html>.

If more detailed installation information is required please see: *Oracle Fusion Middleware: Installation Guide for Oracle Data Integrator*.

Installing ODI Studio requires administrators rights on the computer it is being installed on. If you want to automatically install patches during the installation process, a My Oracle Support login and password will be required.

---

---

**Note:** Installing and setting up ODI local or standalone agents is covered in a separate chapter of this document: [Installing ODI Agents](#).

---

---

### Installing ODI Studio in a Desktop Environment

The installation for a Desktop Environment is carried out as follows:

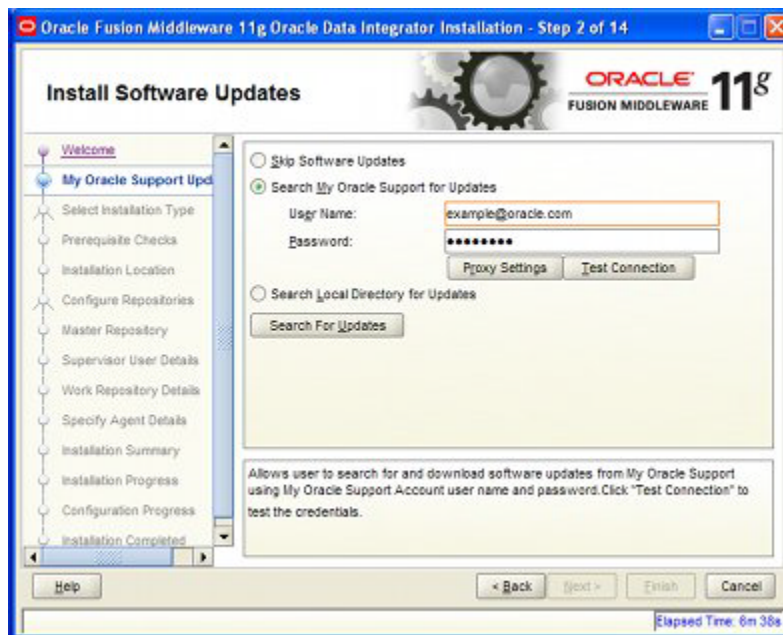
1. Start the installer. The Welcome screen appears.

Figure 9–1 ODI Installation Welcome Screen



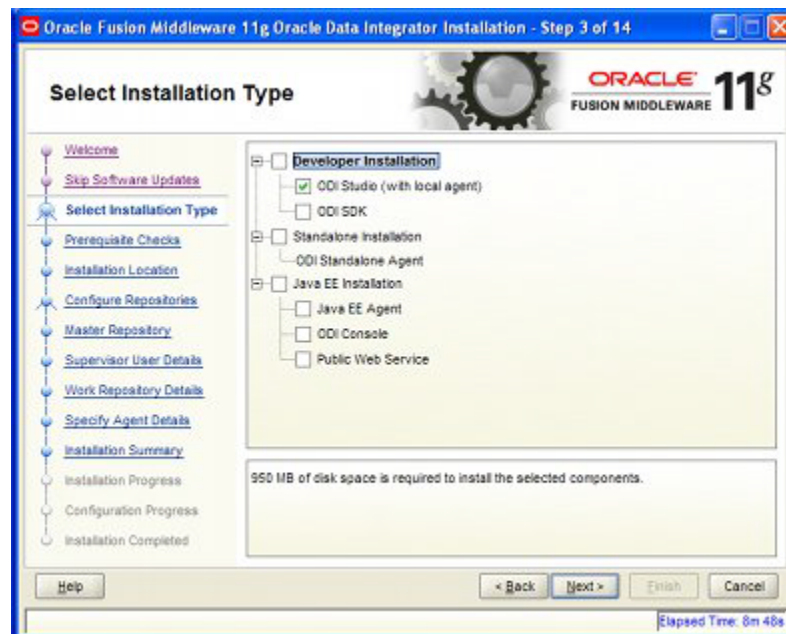
2. Click **Next** to start the installation process. This brings up the Update Screen.

Figure 9–2 ODI Installation - Update Screen



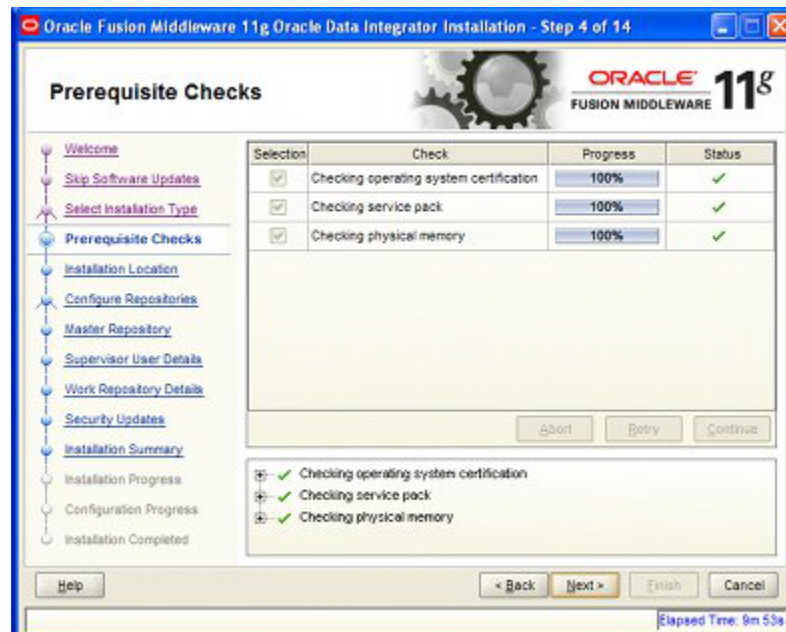
3. Select the required support options. Click **Next** to move to the Install Type screen.

Figure 9–3 ODI Installation - Install Type Screen



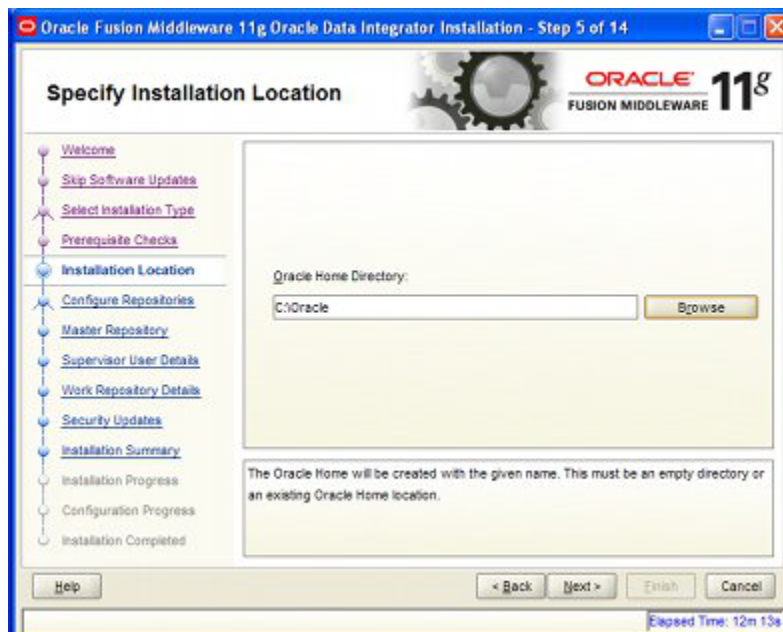
4. Select the **ODI Studio (with local agent)** option. Click **Next** to move to the Prerequisite Checks Screen.

Figure 9–4 ODI Installation - Prerequisite Checks Screen



5. Verify that all checks are satisfactory. Click **Next** to move to the Installation Location Screen.

**Figure 9–5 ODI Installation - Installation Location Screen**



6. Ensure that the rules for assigning Oracle Base and Oracle Home directories are correctly followed.

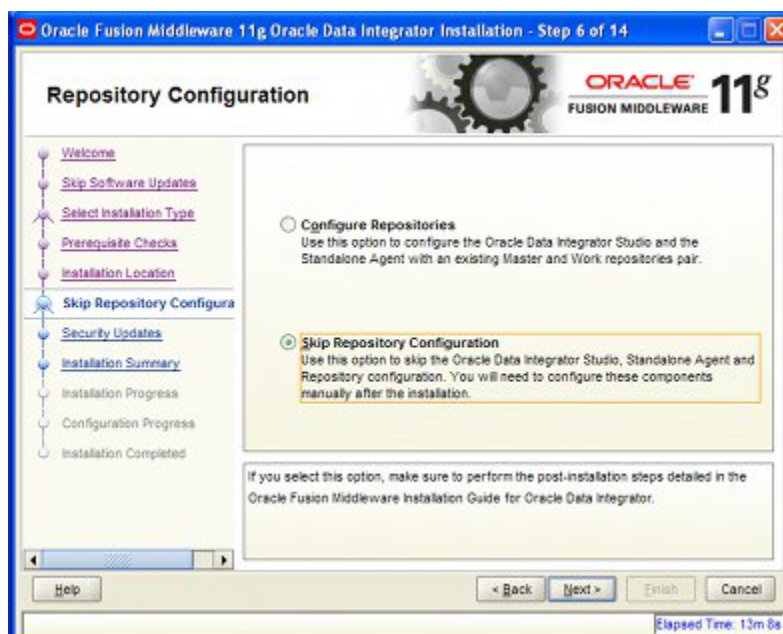
---

**Note:** See the *Oracle Fusion Middleware: Installation Guide for Oracle Data Integrator* for more information on Oracle Home and Oracle Base.

---

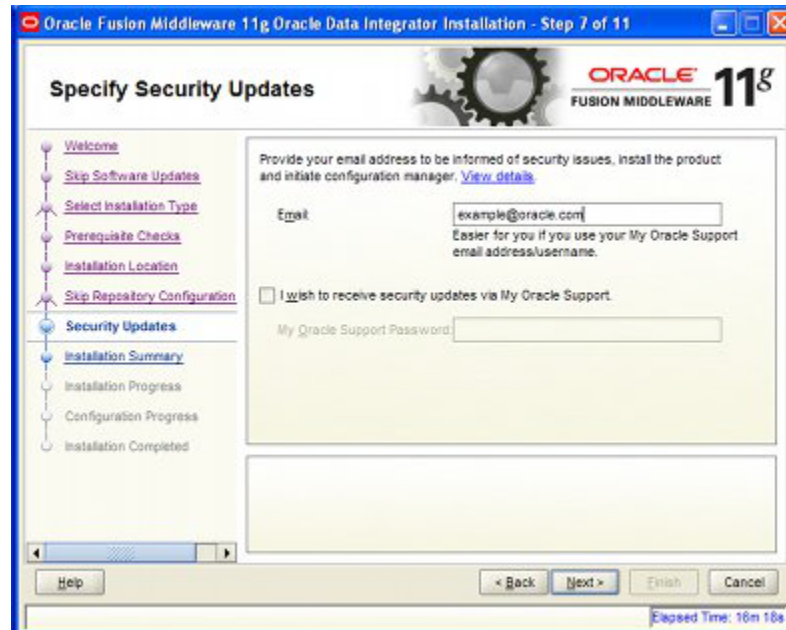
7. Click **Next** to move to the Repository Configuration Screen.

**Figure 9–6 ODI Installation - Repository Configuration Screen**



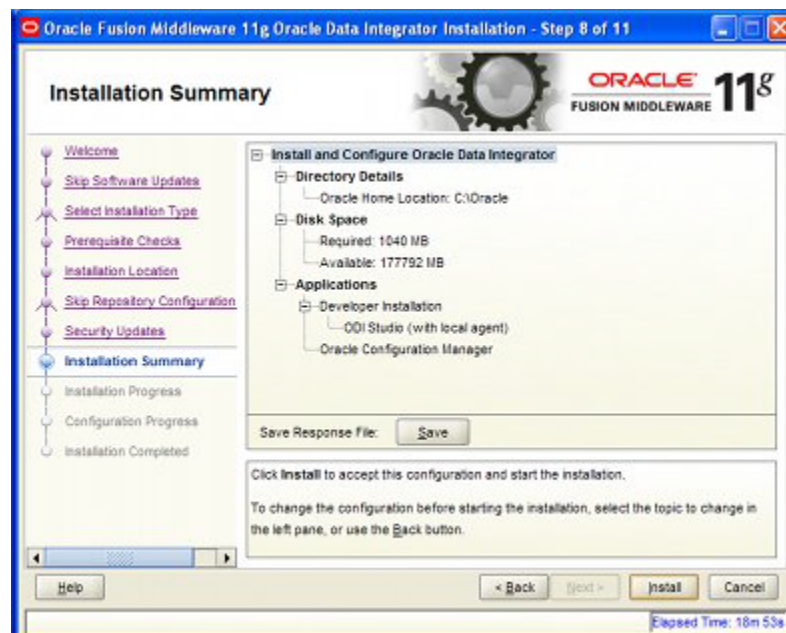
8. Select the **Skip Repository Configuration** radio button. Click **Next**. The Security Updates screen appears.

**Figure 9–7 ODI Installation - Security Updates Screen**



9. Enter the required information concerning Security Updates. Click **Next**. The Installation Summary Screen appears.

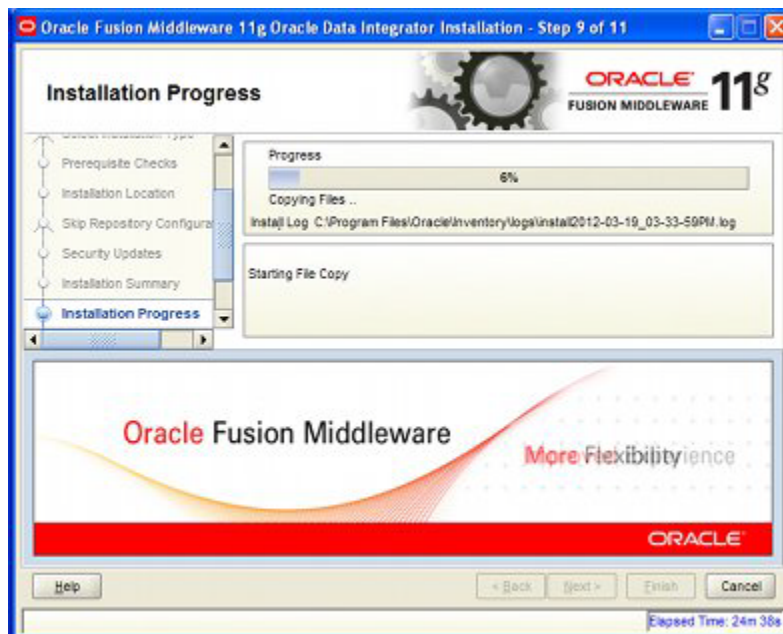
**Figure 9–8**



10. Click **Install**. The Installation Progress screen appears.



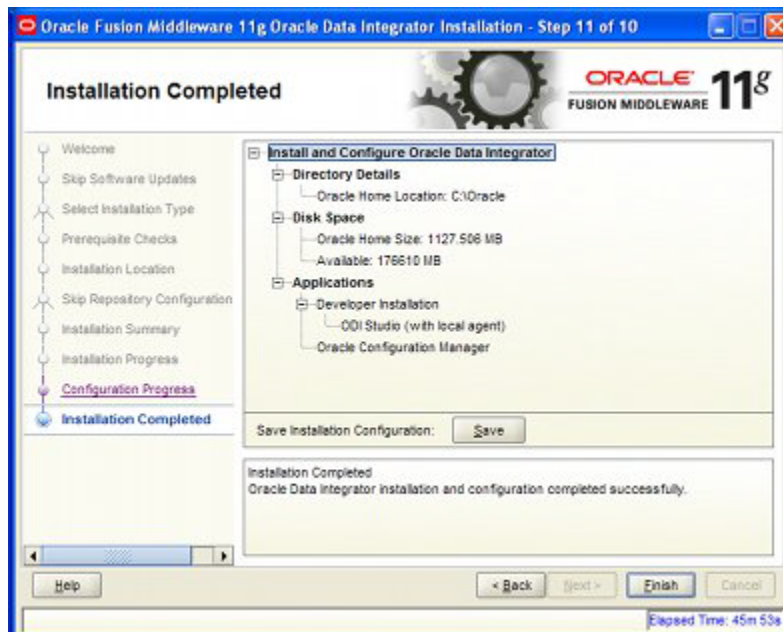
**Figure 9–9 ODI Installation - Installation Progress.**



This screen will show how the installation process is progressing. When it is complete, the **Finish** button becomes active.

11. Click **Finish**. The Installation Completed Screen appears.

**Figure 9–10 ODI Installation - Installation Completed Screen**



On clicking Finish, the screen closes and installation of ODI Studio is complete.

## Reinstalling ODI and Connecting to Existing Repositories

If ODI Studio is deleted and reinstalled, or if ODI Studio is being installed on another computer, it is not necessary to recreate the master and work repositories.

To reconfigure the connections to the repositories, take the following steps:

1. Install or Reinstall ODI.
2. Configure a copy of the **ExtXML.properties** file and place it in the **C:\Oracle\_Base\Oracle Home\oracledi\client\jdev\extensions\oracle.odi.navigator\lib** directory on the computer ODI studio has been installed/reinstalled on.
3. Configure a connection to the **Master Repository**.
4. Configure a connection to the **Work Repository**.

---

---

**Note:** Information on how to configure the ExtXML.properties file, together with the master and work repositories can be found in the chapter on Configuring ODI Studio: [Configuring ODI Studio](#).

---

---

You now have a restored connection to the master and work repositories containing the import information.





---

---

## Installing ODI Agents

This chapter gives basic instructions on how to install an ODI Agent. In order to install this one, you must first download the latest version of ODI from the Oracle Technology Network: <http://www.oracle.com/technetwork/index.html>.

If more detailed installation information is required please see: *Oracle Fusion Middleware: Installation Guide for Oracle Data Integrator*.

Installing an ODI agent requires Administrators rights on the computer it is being installed on. Additionally, if automatically installing patches during the installation process, a My Oracle Support login and password will be required.

---

---

**Note:** Setting up ODI local or standalone agents is covered in a separate chapter of this document.

---

---

ODI agents may be in one of two forms:

- Installed on the same computer as an instance of ODI Studio (local agent).
- Installed by itself on a server (standalone agent).

You can use these agents to execute scheduled imports. Because of latency issues, using an agent installed on the same server as the database instance containing the Macro Space Planning (MSP) database will should result in the fastest import of data.

### Creating the Agent

The agent is installed as follows:

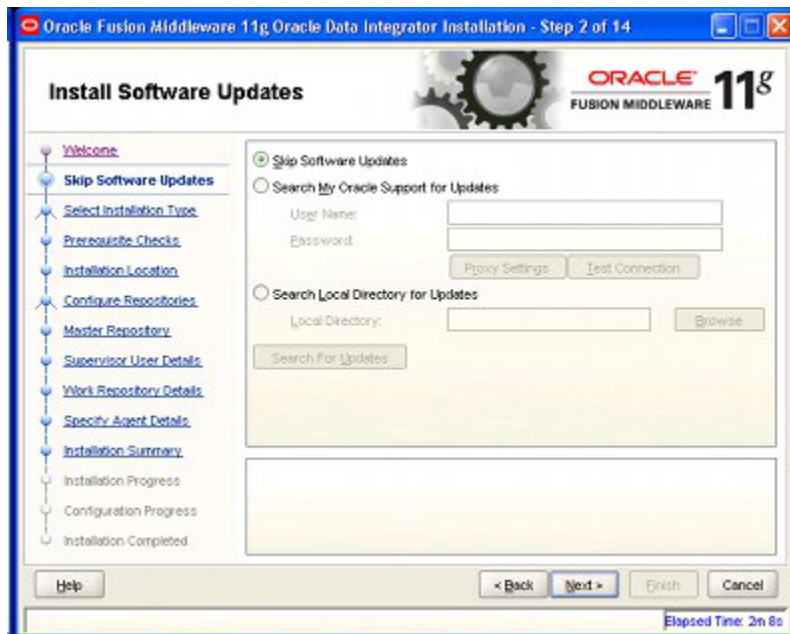
1. Start the Installer. The Welcome screen will appear.

**Figure 10–1 Standalone Agent Install - Welcome Screen**



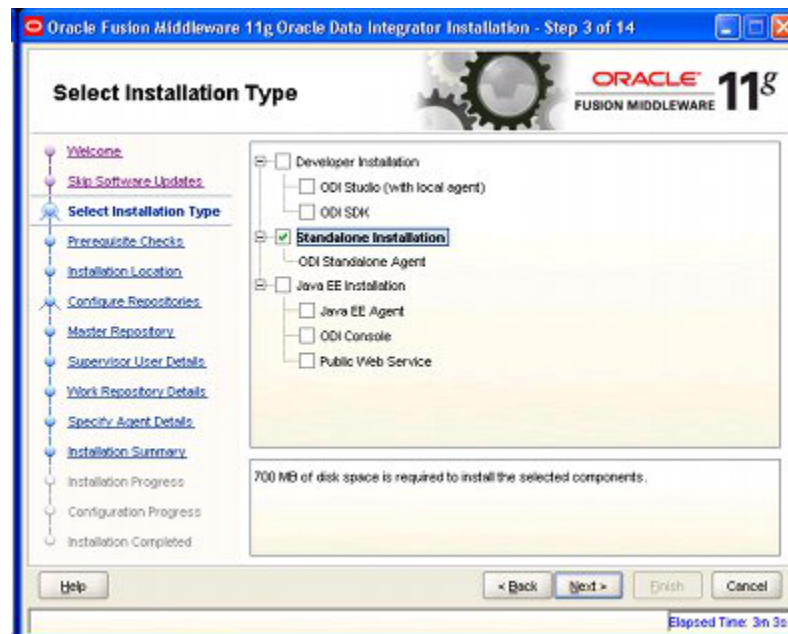
2. Click Next to start the Installation process. The Install Software Updates screen will appear. Enter the information as required.

**Figure 10–2 Standalone Agent Installation - Software Update Screen**



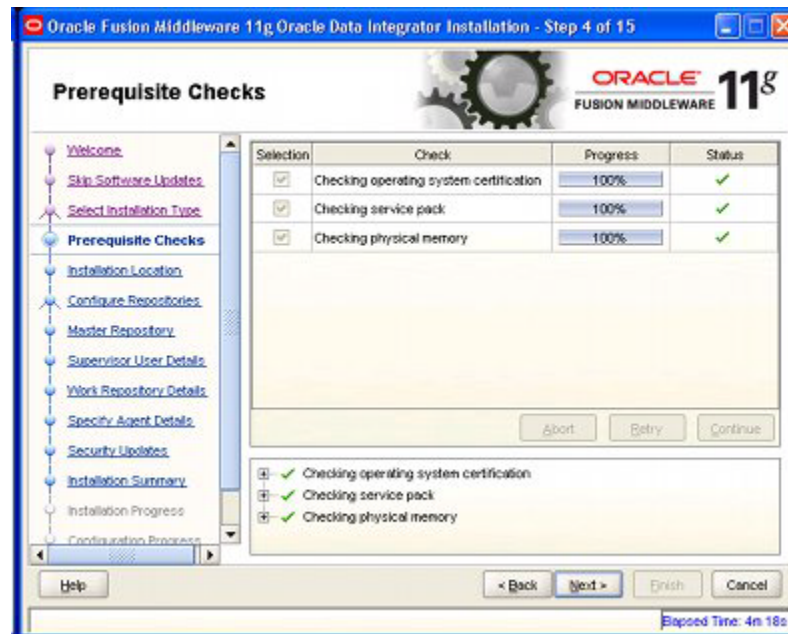
3. Click Next, the Select Installation Type screen will appear.

Figure 10–3 Standalone Agent Installation - Select Type Screen



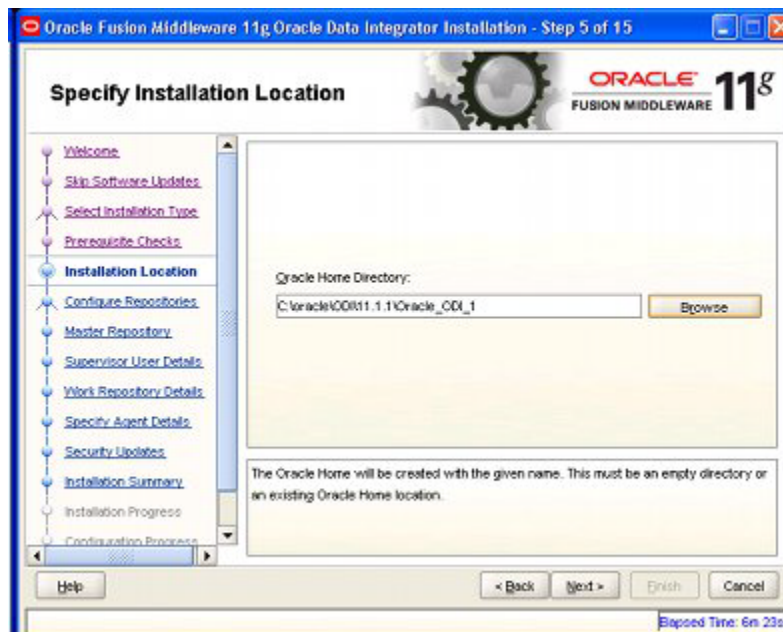
4. Select Standalone Installation and click Next. The Prerequisite Checks screen will appear.

Figure 10–4 Standalone Agent Install - Prerequisite Checks Screen



5. On satisfactory completion of the checks, click Next. The Installation Location Screen will appear.

Figure 10–5 Standalone Agent Install - Installation Location Screen



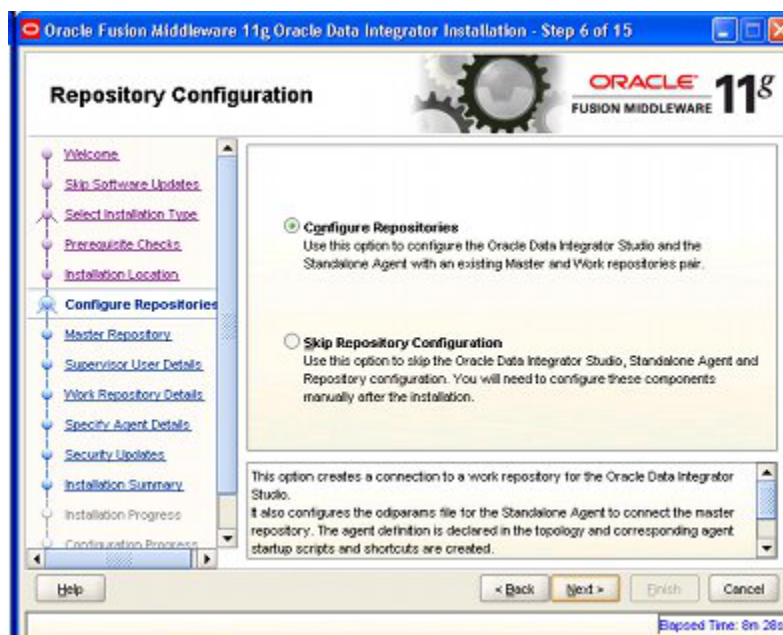
- Specify a suitable Oracle Home directory. Click Next. The Configure Repositories screen will appear.

---

**Note:** It is recommended that you read the ODI documentation to understand how Oracle\_Base and Oracle\_Home directories work.

---

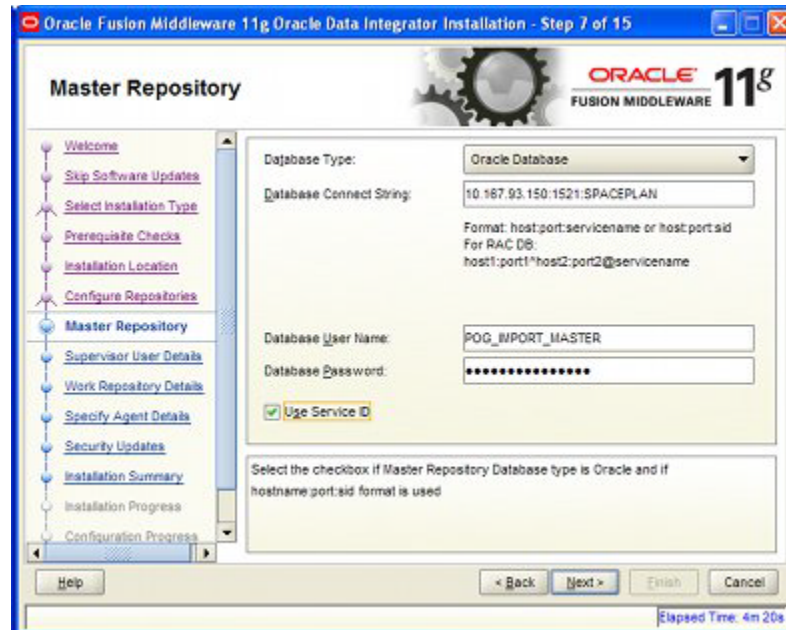
Figure 10–6 Standalone Agent Install - Configure Repositories Screen



Repositories can either be configured now or at a later date. In this example, the repositories will be configured during the installation of the standalone agent.

7. Select the Configure Repositories option and click Next. The Master Repository Screen will appear.

**Figure 10–7 Standalone Agent Install - Master Repository Screen**



This screen is configured as follows:

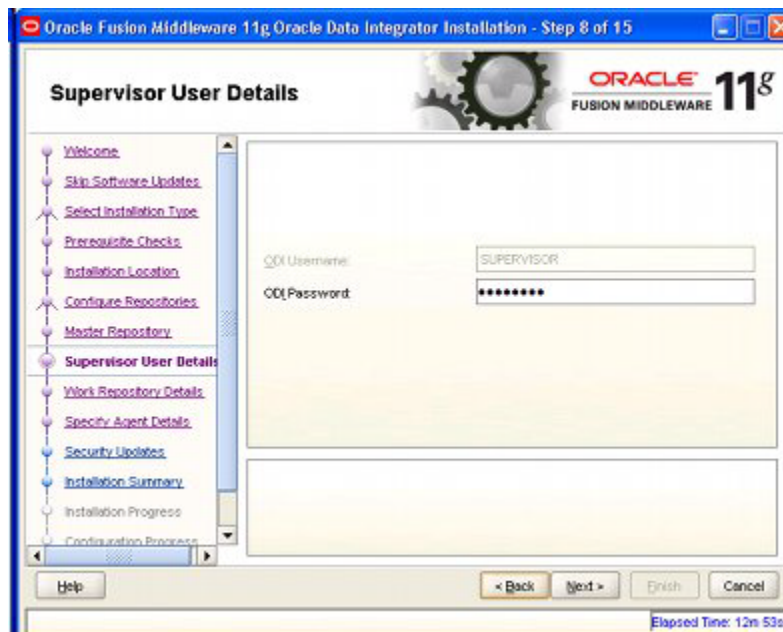
1. Select the database type from the Database Type drop down list. This will be the database instance holding the Master Repository schema. In this example Oracle Database has been chosen.
2. Specify the connection details for the database. The format is:  
`<host>:<port>:<sid>`  
 where the host is the machine name or IP Address, the port is the port for the listener of the database (default is 1521) and the SID is the identifier for the database.
3. Specify the User with access rights to the schema holding the Master Repository. In this example it is **POG\_IMPORT\_MASTER**.
4. Enter the Password for the schema holding the Master Repository. In this case it is **Password\_Master**.

When the details have been entered click Next.

8. The Supervisor User Details screen will appear.



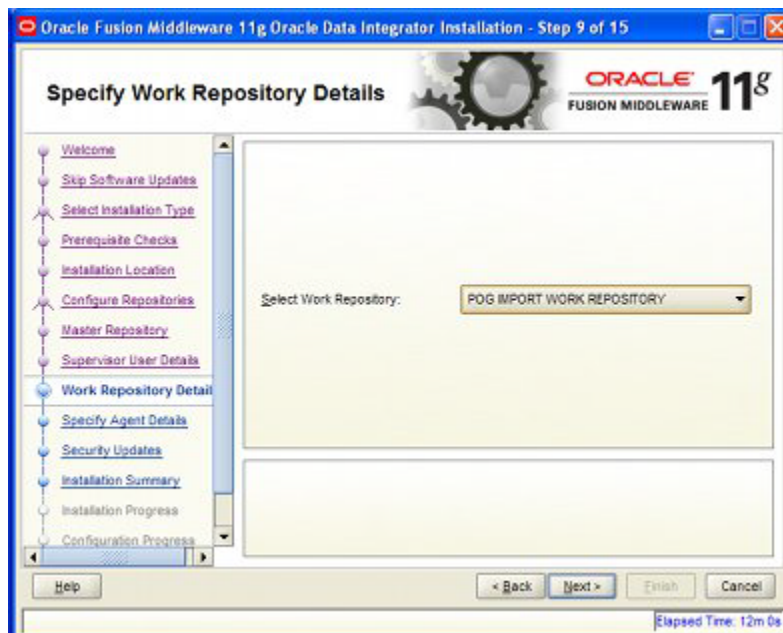
**Figure 10–8 Stand alone Agent Install - Supervisory User Screen**



This screen requires entries for the User Name and Password required to log into ODI. In this instance the default username of **SUPERVISOR** and the default password of **SUNOPSIS** have been used. Click **Next**.

9. The Work Repository Screen appears.

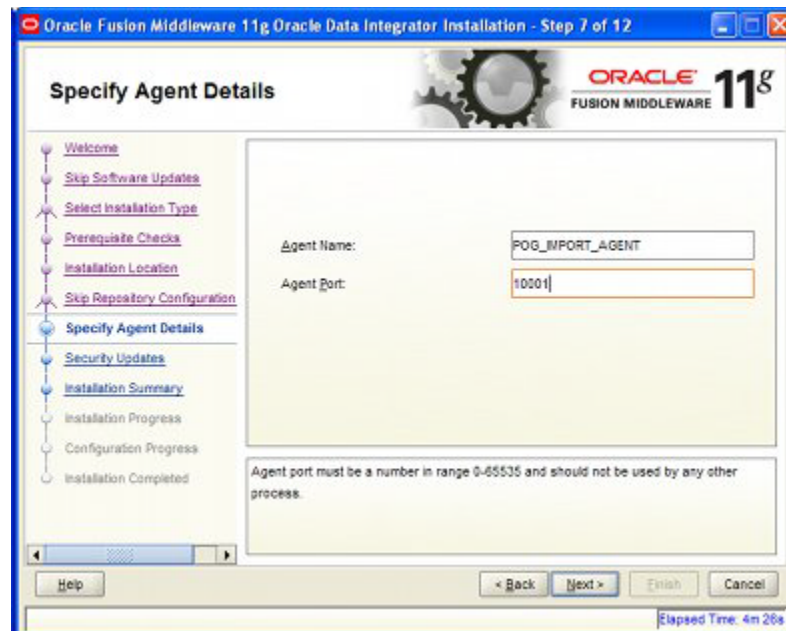
**Figure 10–9 Standalone Agent Install - Work Repository Screen**



Select the required Work Repository from the drop down list. In this example it is **POG IMPORT WORK REPOSITORY**. Click **Next**.

10. The specify Agent Details Screen will appear.

**Figure 10–10 Standalone Agent Install - Agent Details Screen**



Specify the following information:

- **Agent Name:** the name the agent will be identified by when it is created. In this example it is called POG\_IMPORT\_AGENT.
- **Agent Port:** the port the standalone agent will communicate through.

---

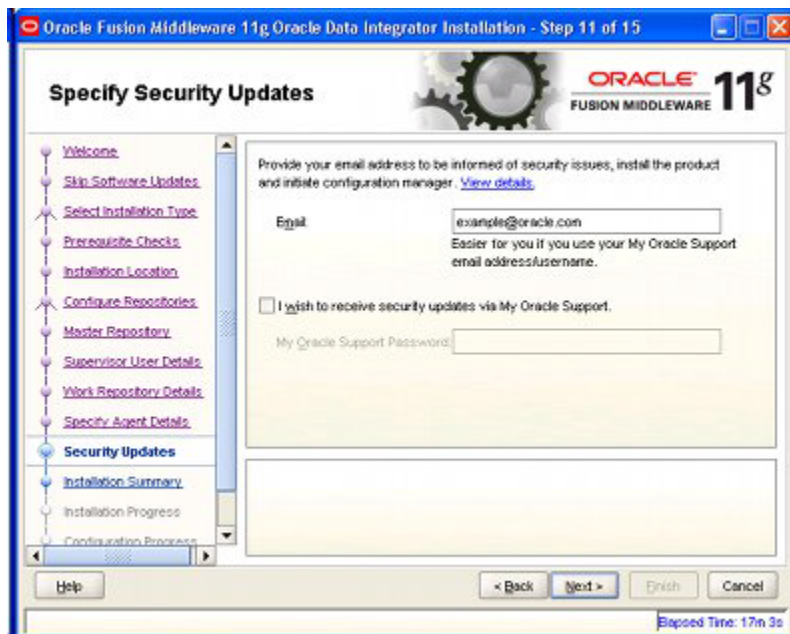
**Note:** It is recommended that implementers check with the System Administrator for the computer the Standalone Agent is being installed on to ensure they are assigning an unused port.

---

When the information is configured, click **Next**.

11. The Security Updates Screen will appear.

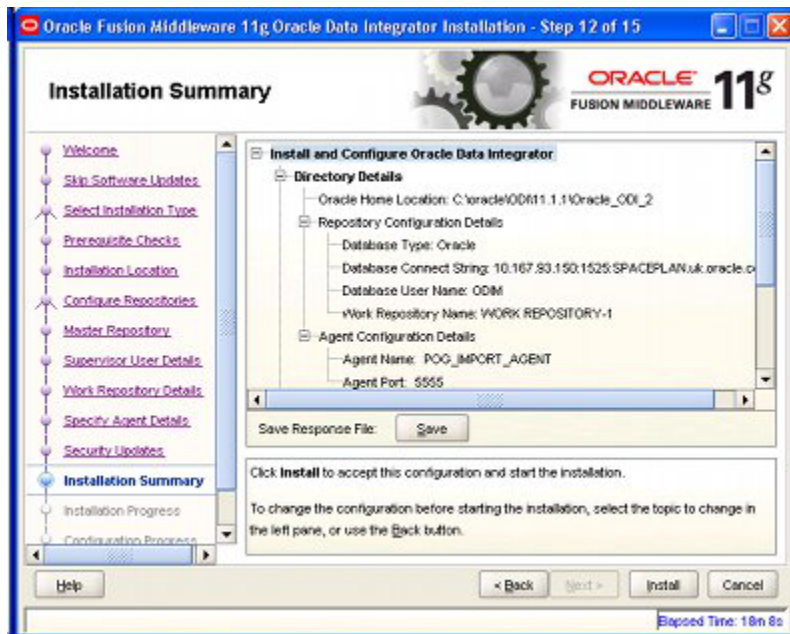
**Figure 10–11 Standalone Agent Install - Security Updates Screen**



Enter the security update details and click **Next**.

12. The Installation Summary screen will appear.

**Figure 10–12 Standalone Agent Install - Installation Summary Screen**

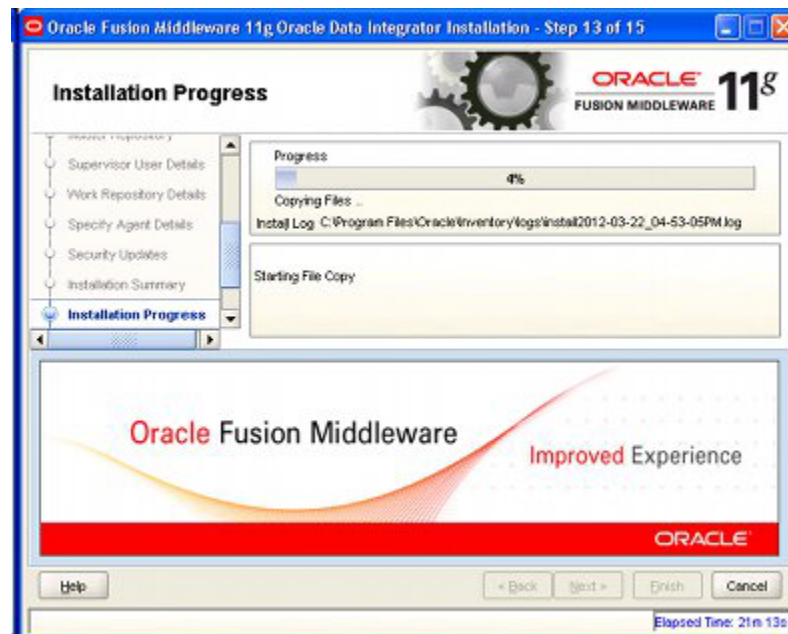


If the details in the Installation Summary screen are correct, click **Install**.

13. The Installation Progress Screen will appear.

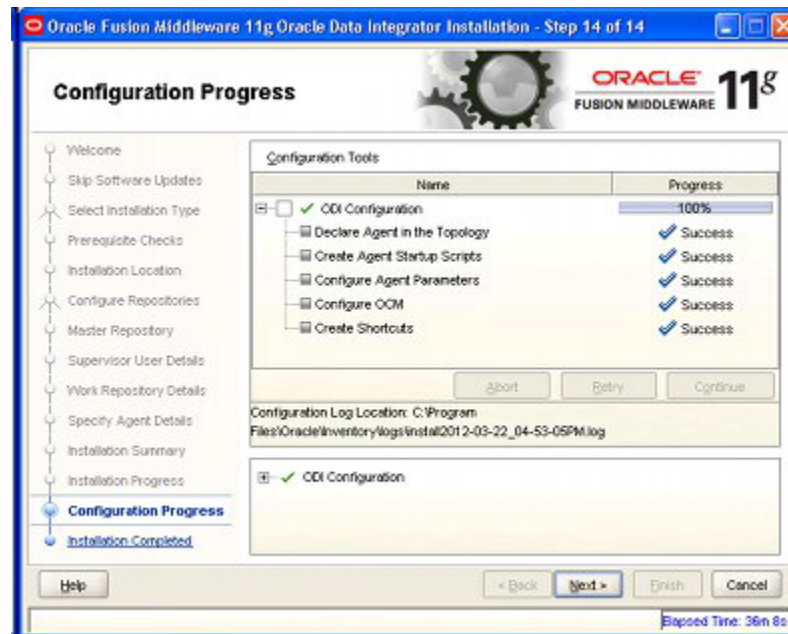


**Figure 10–13 Standalone Agent Install - Installation Progress Screen**



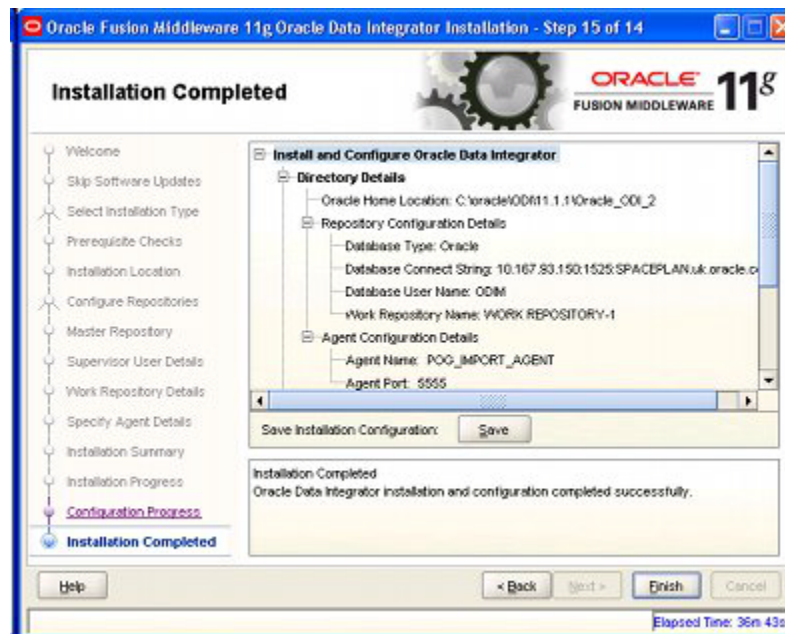
On completion of installation, the Configuration Progress Screen will appear. No action is required for this screen.

**Figure 10–14 Standalone Agent Install - Configuration Progress Screen**



14. When all actions are complete, click Next. The Installation Completed screen will appear.

**Figure 10–15 Standalone Agent Install - Installation Completed Screen**



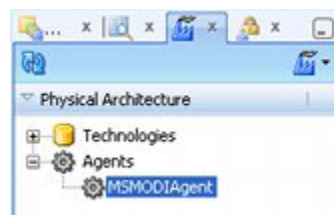
15. Click **Finish** to complete the installation process.

## MSMODIAgent

This agent is imported into ODI Studio when the Master Repository was created. It is recommended that you delete the MSMODIAgent to avoid confusion with any subsequent agents that are created. Take the following steps to delete the MSMODIAgent in ODI Studio after the repositories are created.

1. Select the **Topology Navigator**.
2. Connect to the **Master Repository**.
3. Open the **Physical Architecture** accordion.
4. Highlight the **MSMODIAgent**. Right click and select **Delete**.

**Figure 10–16 Physical Architecture Accordion - MSMODIAgent**



## Starting and Testing the Agent

Take the following steps to start and test the agent:

1. Start the ODI Agent.

2. Open ODI Studio, connect to the Master Repository and test the connection to the Standalone Agent in the **Topology** Navigator.

## Starting the Standalone Agent

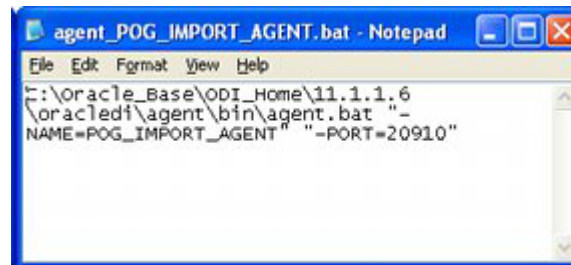
Take the following steps to start the Agent:

1. In order to run the agent, open the command prompt and change the directory path to where the Standalone Agent is present; for example: **C:\Oracle\_Base\Oracle\_Home\oracledi\agent\bin**.

This directory is located on the computer the Standalone Agent was installed on.

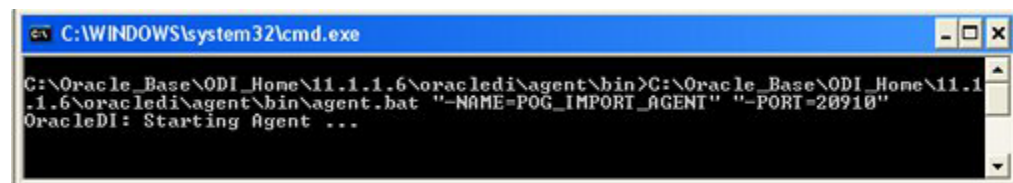
2. This directory will contain a batch (.BAT) file that invokes the agent. It will be named after the agent created in an earlier step. In this example it is **agent\_PLANO\_IMPORT\_AGENT.bat**.

**Figure 10–17 Standalone Agent Batch File**



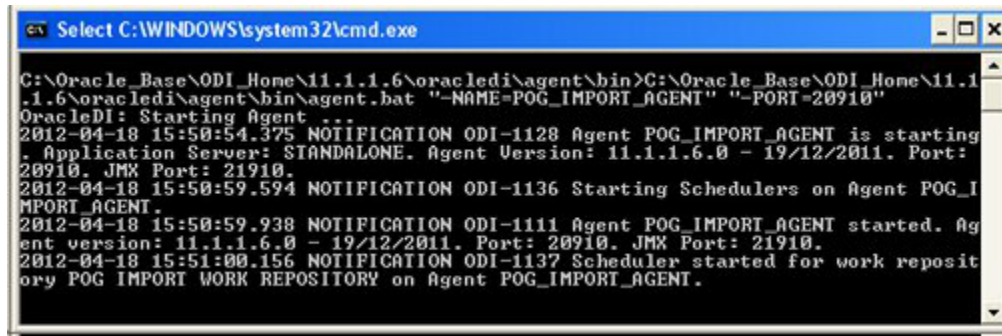
Start the batch file via the command line. A confirmation message will appear in the command line that the agent is starting.

**Figure 10–18 Command Line - Starting Standalone Agent**



After a brief pause, the agent will start; additional confirmation will appear in the Command Line window.

**Figure 10–19 Command Line - Standalone Agent Started**



3. Leave the Standalone Agent running while the connection is tested from ODI Studio.

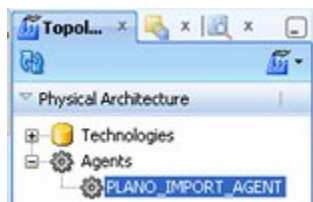
## Testing the Agent from ODI Studio

**Note:** The Master and Work Repositories must first have been set up in order to carry out this test.

The Standalone agent can be tested in ODI Studio as follows:

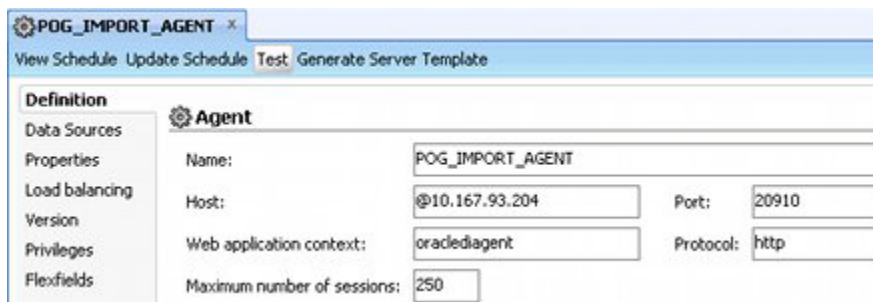
1. Open the **Topology Navigator** and connect to the Master Repository.
2. In the **Physical Architecture** accordion right click on the pertinent agent and select **Open** for more details

**Figure 10–20 Topology Navigator - Available Agents**



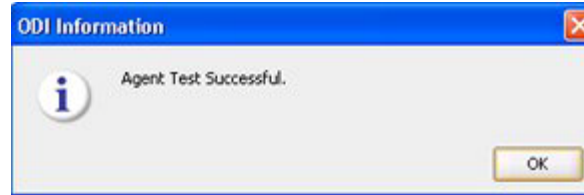
3. The Agent’s dialog box appears. Select the **Definition** Tab.

**Figure 10–21 Physical Agent - Definition Tab**



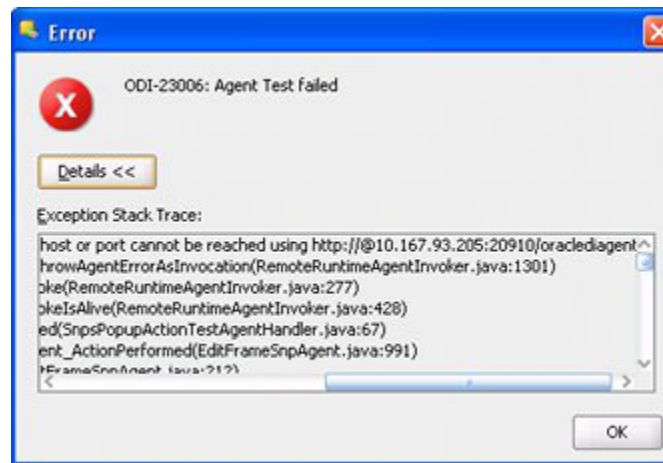
4. Click **Test**. The connection to the Standalone Agent will now be tested. A dialog box reports the results  
 If the connection is successful, it will be reported in a dialog box.

**Figure 10–22 Standalone Agent Connection Confirmation**



5. If the connection fails, an error message will result.

**Figure 10–23 Standalone Agent Connection Error Dialog Box**



### Sources of Connection Errors

Sources of potential connection errors are:

- The Standalone Agent has not started correctly.
- The connection details in the **Definition** tab of the physical agent in ODI Studio are incorrect.
- One or more parameters in the **odiparams.bat** file are incorrect.

## ODI Agents and Scheduling

Once the planogram import process is ready to go live, it will probably be set up to execute on a regular schedule. This can be done in two main ways:

- Using an agent and ODI's internal scheduling tool.
- Starting and stopping an agent by means of an external scheduling tool.

This is discussed in more detail in a later section: [Running ODI Planogram Imports](#).



---

---

## Configuring ODI Studio

This chapter describes how to configure ODI Studio.

---

---

**Note:** Setting up the ODI local or standalone agent is covered in a separate chapter of this document.

---

---

### Configuring Access to the ODI Staging Area

#### Purpose

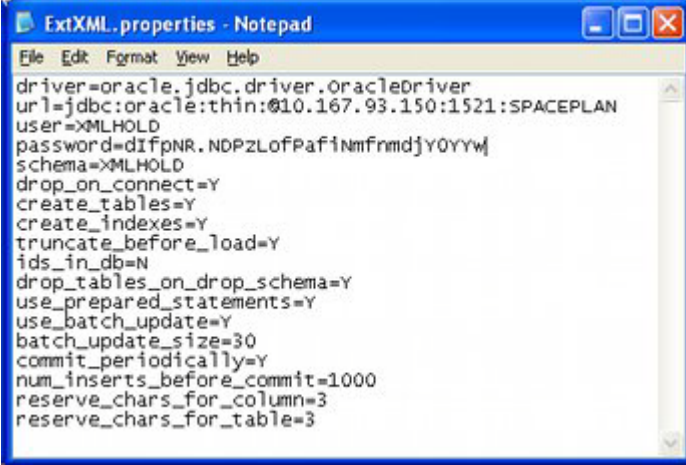
The ODI scenarios for planogram import use the ODI Staging Area to temporarily hold information while executing the import. ODI references the **ExtXML.properties** file for information on how to access the staging area.

#### Configuring and Placing the File

The ODI Staging Area is a schema where data is held temporarily prior to import in the MSP database. Data for the connection is held in a file called **ExtXML.properties**. For ODI Studio, a copy of the file must exist in **C:\Oracle\_Base\Oracle\_Home\oracledi\client\jdev\extensions\oracle.odi.navigator\lib** directory.

A copy of the **ExtXML.properties** file is supplied with the installation package. It can be found in the **External DB Properties** folder.

**Figure 11–1** *ExtXML.Properties File*



```
ExtXML.properties - Notepad
File Edit Format View Help
driver=oracle.jdbc.driver.OracleDriver
url=jdbc:oracle:thin:@10.167.93.150:1521:SPACEPLAN
user=>XMLHOLD
password=dIfpNR.NDPzLoFPafinmfrmdjY0YYW}
schema=>XMLHOLD
drop_on_connect=Y
create_tables=Y
create_indexes=Y
truncate_before_load=Y
ids_in_db=N
drop_tables_on_drop_schema=Y
use_prepared_statements=Y
use_batch_update=Y
batch_update_size=30
commit_periodically=Y
num_inserts_before_commit=1000
reserve_chars_for_column=3
reserve_chars_for_table=3
```



Take the following steps to update the appropriate fields in the ExtXML.properties file.

1. Change the URL to that of the server holding the database instance with the Staging Area schema. This is of the form:  

```
jdbc:oracle:thin:@<host>:<port>:<sid>
```

  - **Host** is the IP Address or Computer Name the database instance is hosted on. this will generally be a **Static IP Address**.
  - **Port** is the port the database listener is configured for. The default is **1521**.
  - **SID** is the Oracle SID for the database. In this example, it is **SPACEPLAN**.
2. Set the User to the user who owns the schema. In this example, this is **XMLHOLD**.
3. Set the Schema name to that of the schema. In this example, it is **XMLHOLD**.
4. Set the Password to the encoded form for that of the schema. In this example it is **Password\_XMLHOLD**. (See below for how to do this).
5. Save the modified file.
6. Place a copy of the modified file in the required directory.

## Encoding the Password

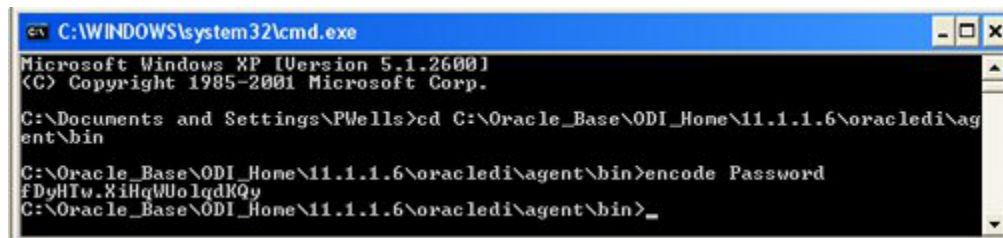
The ExtXML.properties file requires an encrypted version of the password for the staging area. This can be generated by using the Encode.bat file found in the C:\Oracle\_Base\Oracle\_Home\oracledi\agent\bin directory for the ODI Standalone Agent.

---

**Note:** A copy of the ODI agent must be installed to access the **Encode.bat** file. This file is not installed with ODI Studio itself.

---

**Figure 11-2 Command Line - Encode Password**



To encode the password:

1. Open a Command Line prompt.
2. Change to the C:\Oracle\_Base\Oracle\_Home\oracledi\agent\bin directory.
3. Use the **encode <password>** command. In the above example, the word **Password** is being encoded.
4. Paste the encoded password into the **ExtXML.properties** file.
5. Save the modified file.



---

---

**Note:** The encode.bat file will generate alternative versions of the encrypted password each time it is run.

---

---

## Create and Configure ODI Master Repository

You install ODI and configure the ODI Master Repository. This holds information on the objects that ODI can connect to.

### Prerequisites

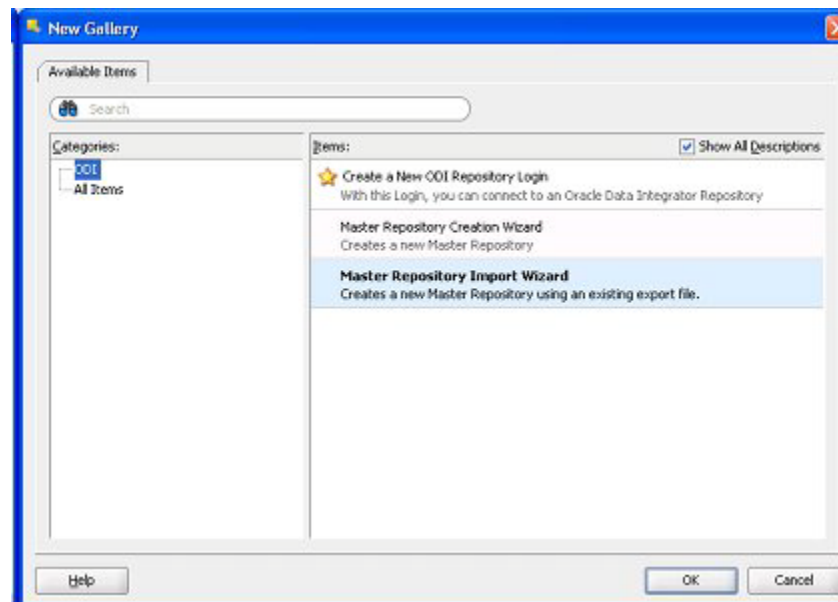
A schema to hold the ODI Master Repository must first have been created in a database instance. In addition, the **ExtXML.Properties** file must have been configured and a copy placed in the **C:\Oracle\_Base\Oracle Home\11.1.1\Oracle\_ODI\_1\oracledi\client\jdev\extensions\oracle.odi.navigator\lib** directory. You must also have knowledge of the pertinent users and passwords.

## Create Master Repository

Take the following steps to create the Master Repository:

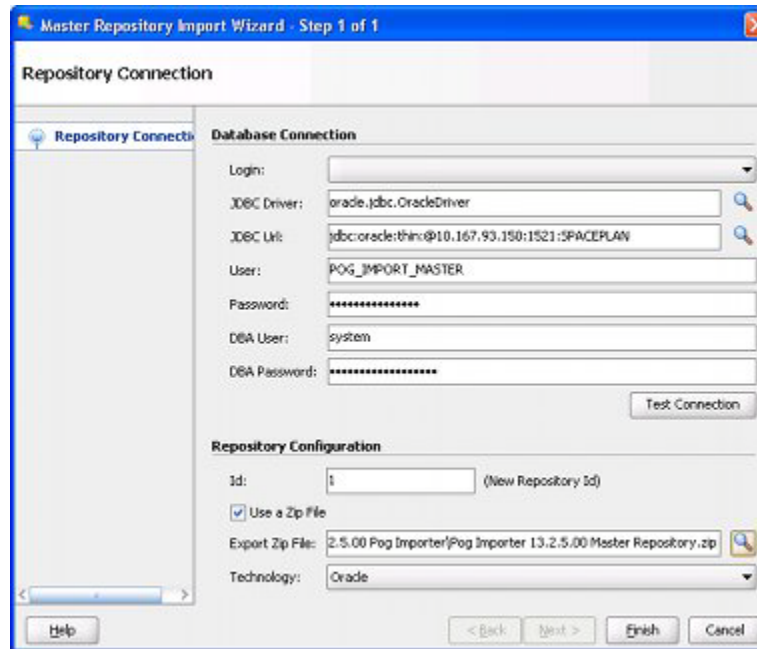
1. Open **New Gallery** by choosing **File > New**. The New Gallery dialog box opens.

*Figure 11-3 ODI New Gallery Dialog Box*



2. Select **Master Repository Import Wizard**. This will bring up the Wizard.

Figure 11–4 Master Repository Import Wizard dialog box



This dialog box is configured as follows:

- a. The Login is blank at this stage as the Master Repository has not been configured.
- b. The JDBC Driver is set to the appropriate option for the database type. It can be selected by clicking the magnifying glass icon to the right of the text field. In this case, it needs to be set for an Oracle database.
- c. The JDBC URL is set as appropriate. For an Oracle database the format is:  
`jdbc:oracle:thin:@<host>:<port>:<sid>`

where the host is the machine name or IP Address, the port is the port for the listener of the database (default is 1521) and the SID is the identifier for the database. The format for different database types can be obtained by clicking the magnifying glass icon to the right of the text field.

In this example the JDBC URL is set for the SPACEPLAN database instance as this holds the specified schema.

---

**Note:** Again, this IP Address should be a Static IP Address.

---

- d. The User is set to that for the schema owner for the Master Repository. In this example the Schema owner is **POG\_IMPORT\_MASTER**.
- e. The Password is set to that for the schema owner for the Master Repository. In this example, it is **Password\_Master**.
- f. The DBA User is set to the Login ID for the DBA for the database instance hosting the Master Repository. In this example, it is **system**.

- g. The DBA Password is set to the Password for the DBA for the database instance hosting the Master Repository. In this example, it is **Password\_Spaceplan**.
- h. Click **Test Connection** to verify the log in details are correct.
- i. The ID is a numerical identifier for the repository being created. This needs be a unique ID not used by any other repository.

---

**Note:** The ID should not be set to 1. This value is imported when the Master Repository is created. ODI then generates an error message stating that the Master Repository already exists.

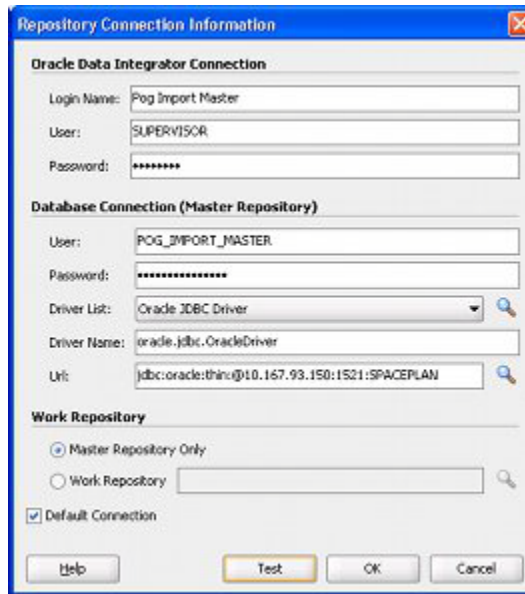
---

- j. The **Use a Zip File** check box should be selected.
- k. The path to the zip file required can be selected by clicking the magnifying glass icon to the right of the text field and browsing to the required zip file and directory.  
  
The Zip File to be used is supplied with the functionality and is called **Pog Importer 13.2.5.00 Master Repository.zip**. It can be found in the unzipped **ODI 13.2.5.00 Pog Importer** folder. It contains files required to configure the repository.
- l. The Technology type for the database is set using the drop down list. In this example, it has been set to **Oracle**.
- m. Click **Finish** to create the Master Repository.

## Creating a Login for the Master Repository

The next stage is to create a Log in for the Master Repository. This is done as follows:

1. Open New Gallery by choosing **File > New**. The New Gallery dialog box will open.
2. Select the **Create a New Repository Login** option. Click **OK**.
3. The Repository Connection Information dialog box appears.

**Figure 11–5 Master Repository - Connection Configuration dialog box**

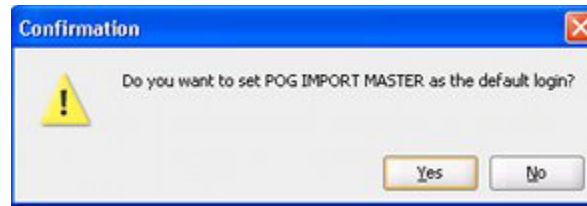
4. The dialog box is configured as follows:
  - a. The Login Name is set to a suitable identifier for this log in. In this case, it has been set to Pog Import Master.
  - b. The User for the ODI Connection is set to the system login for ODI. In this case, the default ODI user of **SUPERVISOR** is being used.
  - c. Set the Password for the ODI connection to the system password for ODI. In this case, the default ODI password of **SUNOPSIS** is being used.
  - d. Set the User for the Database Connection to that created for the Master Repository in the database schema. In this case, it is **POG\_IMPORT\_MASTER**.
  - e. Set the Password for the ODI Connection to that created for the Master Repository in the database schema. In this case, it is **Password\_Master**.
  - f. The Driver List (required for connection to the database) is set from the drop down list. In this case, it has been set to Oracle JDBC Driver.
  - g. The Driver Name will populate automatically.
  - h. The JDBC URL is set as appropriate by clicking the magnifying glass icon to the right. The format is:
 

```
jdbc:oracle:thin:@<host>:<port>:<sid>
```

 where the host is the machine name or IP Address, the port is the port for the listener for the database (default is 1521) and the SID is the identifier for the database.  
 The format for different database types can be obtained by clicking the magnifying glass icon to the right of the text field.
  - i. Click **Test** to verify the connection is OK.
  - j. The Work Repository option is set to Master Repository Only using the radio button.

- k. The connect is made the default by checking the Default Connection check box. A confirmatory dialog box will appear. Click **Yes** to confirm

**Figure 11–6 Default Login Confirmation dialog Box.**



- l. Click OK to create the connection to the Master Repository.

## Configure Topology - Physical Architecture

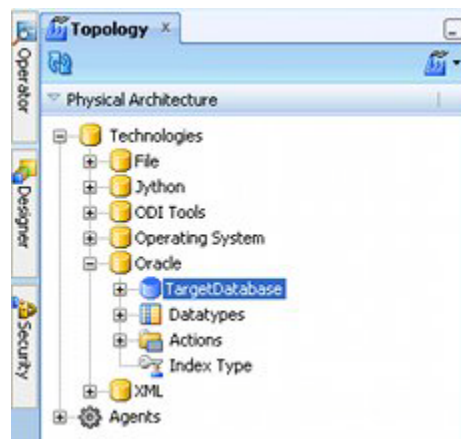
This section explains how to configure the physical architecture in the ODI **Topology** navigator.

### Target Database

The target database is where the planogram data is imported into. This will be the Macro Space Planning database. Take the following steps to enter the details:

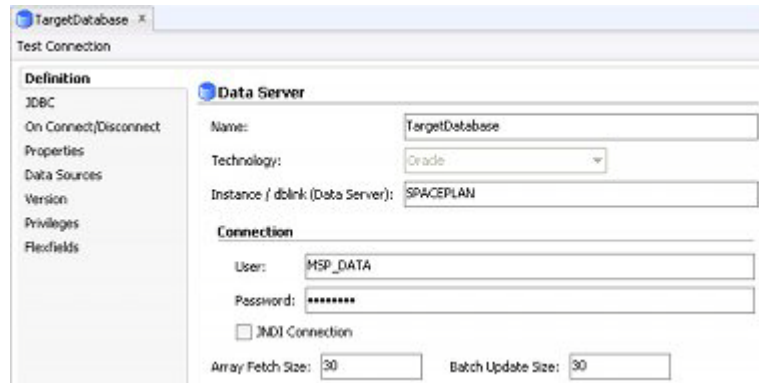
1. Within ODI, select the **Topology** navigator.
2. Select the **Physical Architecture** option by clicking on the small triangle to the left of the text. The list of options will open. Expand the **Technologies** option and select the **Oracle** option, followed by **Target Database**.

**Figure 11–7 Topology Tab - Physical Architecture Option**



3. Right Click on Target Database and select Open. The Target Database tab will appear.
4. Select the Definition sub-tab.

**Figure 11–8 Target Database - Definition Option**



Configure the options as follows:

- a. Enter the name of the database in the Instance field. This is the physical name of the data server. In this example, it is **SPACEPLAN**.
- b. Enter the User: this should be the user for the Macro Space Planning database. In this case, it is **MSP\_DATA**.
- c. Enter the Password: this should be the password for the Macro Space Planning database. In this example it is **Password\_MSP\_DATA**.

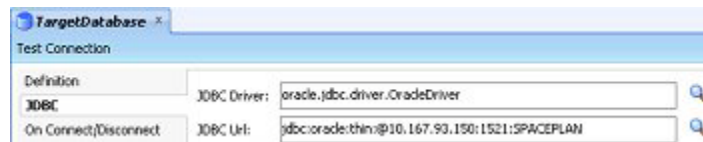
---

**Note:** The access details for the Macro Space Planning database should confer select, insert and object creation privileges.

---

5. Select the **JDBC** sub-tab.

**Figure 11–9 JDBC Sub-Tab**



Configure the Options as follows:

- a. The **Driver List** (required for connection to the database) is set from the drop down list by clicking the magnifying glass icon to the right. In this case it has been set to Oracle JDBC Driver.
- b. The **JDBC URL** is set as appropriate by clicking the magnifying glass icon to the right. The format is:

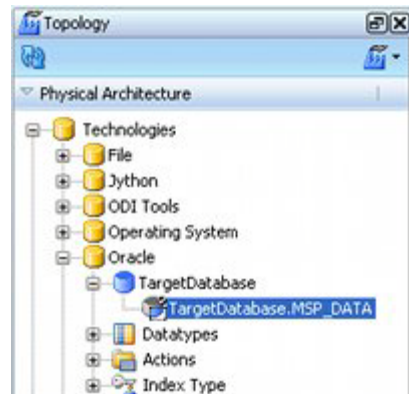
jdbc:oracle:thin:@<host>:<port>:<sid>

where the host is the machine name or IP Address, the port is the port for the listener for the database (default is 1521) and the SID is the identifier for the database.

The format for different database types can be obtained by clicking the magnifying glass icon to the right of the text field.

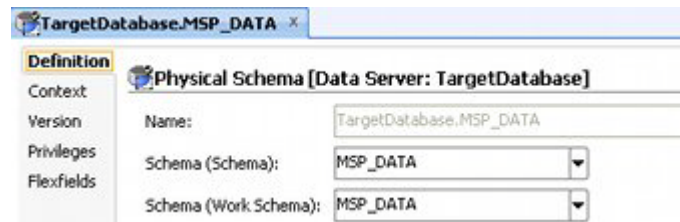
- Further expand the Database hierarchy to show the schema. Right click and select **Open**.

**Figure 11–10 Physical Architecture - Database Schema.**



- The Target Database Physical Schema tab will appear. Set the Schema and Work Schema details to those of the schema holding the Macro Space Planning database. In this example, both values have been set to **MSP\_DATA**.

**Figure 11–11 Target Database Physical Schema tab.**



- Click the Save icon on the ODI Studio toolbar to save the changes.

## XML Technology

---

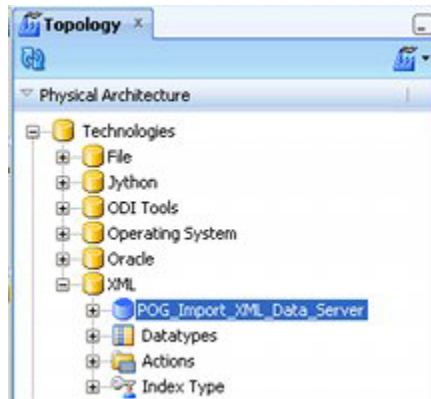
**Note:** Before configuring the XML physical architecture, a copy of the correctly configured **ExtXML.Properties** file must be in the **Oracle\_Base/Oracle\_Home/Version/oracledi/client/jdev/extensions/oracle.odi.navigator.lib** directory.

---

The XML Technology option is used to store the path and file name of the XSD file used to validate the form of the XML being presented for import.

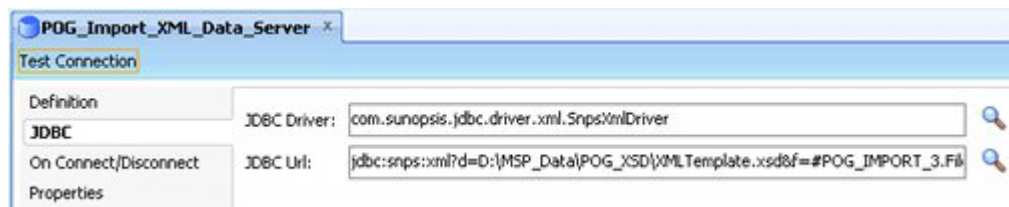
- Within ODI select the **Topology** navigator.
- Select the **Physical Architecture** option by clicking on the small triangle to the left of the text. The list of options opens. Expand the hierarchy.

**Figure 11–12 Physical Architecture - XML Option**



3. Select the **POG\_Import\_XML\_Data\_Server** option, right click and select Open.
4. This will bring up the **POG\_IMPORT\_Data\_Server** tab. Select the **JDBC** option.

**Figure 11–13 POG\_Import\_XML\_Data\_Server Tab**



5. Click the magnifying glass to the right of the JDBC Driver text box. Ensure the Driver Name is set to **ODI JDBC Driver for XML**.
6. Click the magnifying glass to the right of the JDBC text box. The format of the text string is `jdbc:snps:xml?f=<filename>[&s=<schema>&<property>=<value>...]`

Items of data that need to be entered areas follows:

- a. The 'd' part of the JDBC URL is the path to and file name of the XSD file. In this example the path is the local drive `D:\MSP_Data\POG_XSD` that was created earlier. The file is `XMLTemplate.xsd` that was supplied with the functionality.

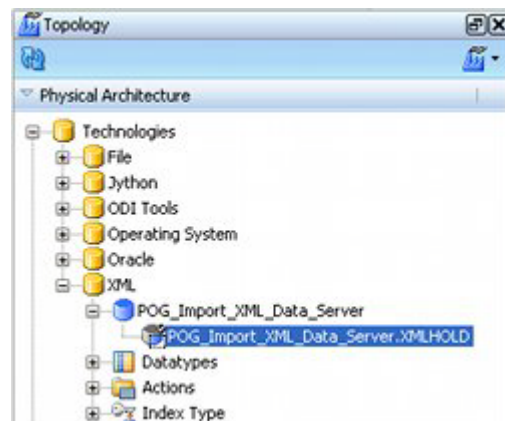
---

**Note:** For anything other than a local drive the full path to the XSD file must be specified. An example would be `\\10.167.93.150\MSP_Data\POG_XSD`.

---

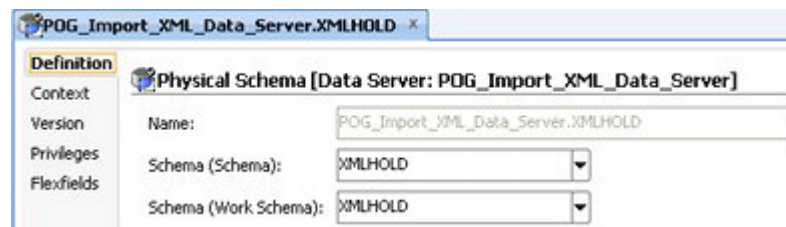
- b. The 's' part of the JDBC URL is the name of the schema the XSD will be used to create temporary tables in. In this case, this is the name of the ODI Staging area defined earlier - this is **XMLHOLD**.
7. Further expand the XML hierarchy to show the Schema required for the XML data Server. Right click and select Open.



**Figure 11–14 Physical Architecture - XML Schema Option**

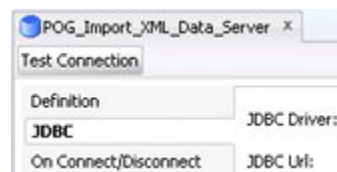
8. The Physical Schema tab will appear. Edit the **Schema (Schema)** and **Schema (Work Schema)** fields so that they hold the name of the Schema previously created in the Database instance.

In this example the Schema Name is **XMLHOLD**.

**Figure 11–15 XML Schema - Physical Schema Tab**

### Testing the Physical Architecture settings for the XML Schema

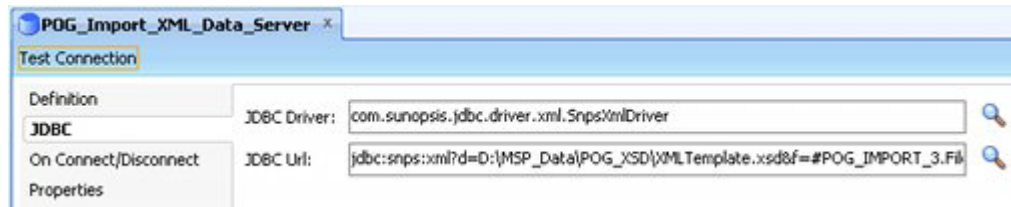
It is possible to test the connection to the XML schema using the **Test Connection** button on the POG\_Import\_XML\_Data\_Server tab.

**Figure 11–16 POG\_Import\_XML\_Data\_Server Test Button**

However, because of the way ODI Studio is configured it is not possible to test the connection with the full text string present. Instead, the following procedure must be used.

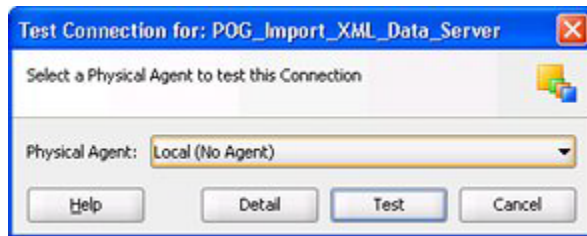
1. Open the POG\_Import\_XML\_Data\_Server tab. Save a copy of the Text string in the JDBC Url field into a text file.

**Figure 11–17 POG\_Import\_XML\_Data\_Server - JDBC Url field**



2. Delete the text `&f=#POG_IMPORT_3.FileName` from the JDBC Url field.
3. Click the **Test** connection button. After confirming you wish to save the changes, the Test Connection dialog box will appear.

**Figure 11–18 Test Connection Dialog Box**



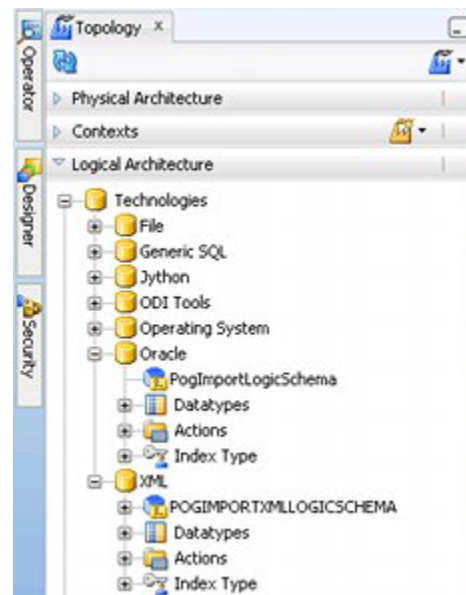
4. Click **Test**. If the connection information is correct, a confirmatory dialog box appears.
5. Delete the edited text string from the **JDBC Url** field and paste in the saved original version from the text file.
6. Save the changes using the **Save** icon on the ODI Studio toolbar.

## Verify Topology - Logical Architecture

These values should have been set when the master repository was imported. The values can be confirmed as follows:

1. Within ODI, select the **Topology** navigator.
2. Select the **Logical Architecture** option by clicking on the small triangle to the left of the text. The list of options opens. Expand the hierarchy.

Figure 11–19 Topology Tab - Logical Architecture



3. The option under Oracle should be **PogImportLogicSchema**.
4. The option under XML should be **POGIMPORTXMLLOGICSHEMA**.

## Creating the Work Repository

The Work Repository holds information on how data can be manipulated.

### Prerequisites

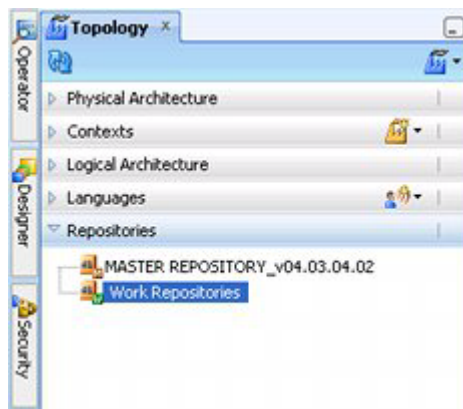
A schema to hold the ODI Work Repository must first have been created in a database instance.

## Creating the Work Repository

Take the following steps to create the Work Repository

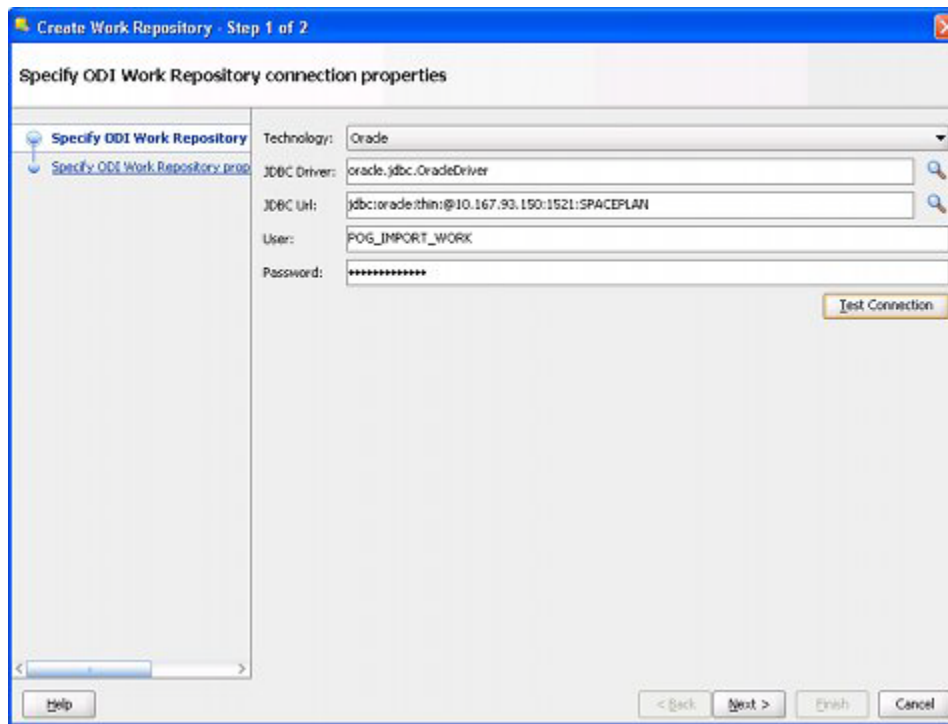
1. Within ODI, select the **Topology** navigator.
2. Select the **Repositories** option by clicking on the small triangle to the left of the text. The list of options opens. Expand the hierarchy.

**Figure 11–20 Topology Tab - Repository Option**



3. Select Work Repositories. From the right click menu, select **New Work Repository**.
4. The first page of the Create Work Repository dialog box appears.

**Figure 11–21 Create Work Repository dialog box - Page 1**



The Screen is configured as follows:

- a. The **Technology** is set from the drop down list. In this example it is Oracle.
- b. The **Driver List** (required for connection to the database) is set from the drop down list by clicking the magnifying glass icon to the right. In this case it has been set to Oracle JDBC Driver.
- c. The **JDBC URL** is set as appropriate by clicking the magnifying glass icon to the right. The format is:

`jdbc:oracle:thin:@<host>:<port>:<sid>`

where the host is the machine name or IP Address, the port is the port for the listener for the database (default is 1521) and the SID is the identifier for the database.

The format for different database types can be obtained by clicking the magnifying glass icon to the right of the text field.

- d. The **User** should be the user that was previously configured for the Work Repository schema in the database. In this example, it is **POG\_IMPORT\_WORK**.
  - e. The **Password** is the password for the schema previously created for the work repository. In this case it is **Password\_Work**.
  - f. Click **Test Connection** to verify the connection details are correct.
  - g. Click **Next** to move to the second Screen.
5. The second page of the Create Work Repository dialog box will appear.

**Figure 11–22 Create Work Repository dialog box - Screen 2**

The screen is configured as follows:

- a. Set the **ID** for the Work Repository to a unique value.
- b. Specify the **Name** for the Work Repository.
- c. The Password can be a maximum of ten characters and is used by ODI. It is not associated with any previous passwords and can be defined by the implementers. In this example, it is **Work\_Rep**.

**Note:** This password will be used if the work repository being created is attached to another master repository.

- d. Set the **Work Repository Type** from the drop down list. This will allow the loading of scenarios, etc. into the Work Repository at a later stage. This is set to **Execution**.

**Note:** The Designer Navigator will not be active with work Repositories of type **Execution**. This is not detrimental to executing planogram imports.

- 6. Click **Finish** to save the Work Repository.
- 7. A dialog box will appear asking if you wish to **Create a Login** for the Work Repository. Click on **OK**. The method for doing this is described in the next section.

## Creating the Login for the Work Repository

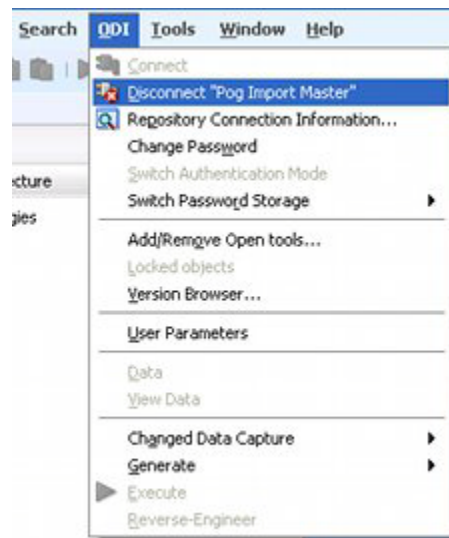
Creating a Login for the Work Repository is a follow on action from creating the Work Repository itself.

### Creating the Work Repository Login Later

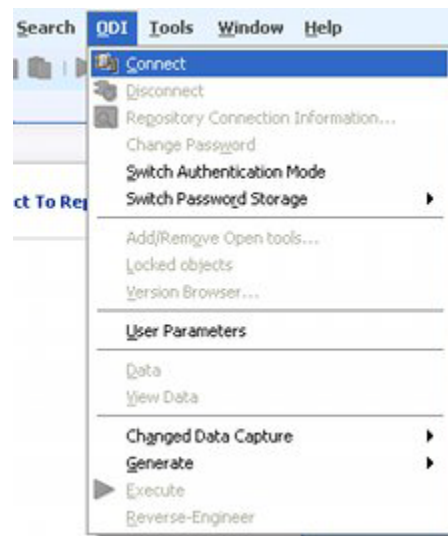
If it is necessary to create the login later (and not as a follow up action to creating the Work Repository), the following actions will bring up the required dialog box.

- 1. If connected to the Master Repository, disconnect using the **Disconnect** option from the **ODI menu**.

**Figure 11–23 ODI Menu - Disconnect Option**



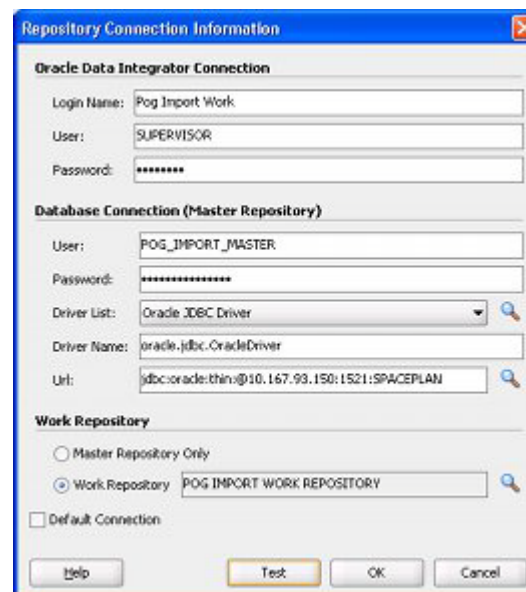
- 2. Select **Connect** from the **ODI menu**.

**Figure 11–24 ODI Menu - Connect Option**

3. The Repository Connection Information dialog box appears. Click the **New** button to open up a blank dialog box.

### Creating the Work Repository Login

Enter data into the Repository connection dialog box to configure the connection to the Work Repository.

**Figure 11–25 Work Repository Connection Information**

The Login can be configured as follows:

1. The Login Name is set to a suitable identifier for this log in. In this case, it has been set to **Pog Import Work**.

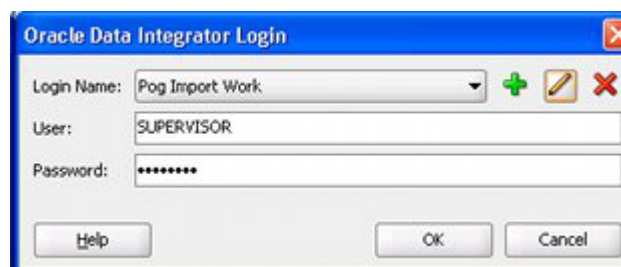
2. The User for the ODI Connection is set to the system login for ODI. In this case the default value of **SUPERVISOR** is being used.
3. Set the Password for the ODI connection to the system password for ODI. In this case the default password of **SUNOPSIS** is being used.
4. Set the User for the Database Connection to that created for the Master Repository in the database schema. In this case, it is **POG\_IMPORT\_MASTER**.
5. Set the Password for the ODI Connection to that created for the Master Repository in the database schema. In this case, it is **Password\_Master**.
6. The Driver List (required for connection to the database) is set from the drop down list. In this case, it has been set to Oracle JDBC Driver.
7. The Driver Name populates automatically.
8. The JDBC URL is set as appropriate. The format is:  

```
jdbc:oracle:thin:@<host>:<port>:<sid>
```

where the host is the machine name or IP Address, the port is the port for the listener for the database (default is 1521), and the SID is the identifier for the database. In this example it is **SPACEPLAN**.

The format for different database types can be obtained by clicking the magnifying glass icon to the right of the text field.
9. Click **Test** to verify the connection.
10. The **Work Repository** option is set to Work Repository using the radio button.
11. The required Work Repository is selected by clicking the magnifying glass icon to the right of the text box and selecting from the list. In this example it is **Pog Import Work**.
12. Click **OK** to create the connection to the Work Repository. The ODI Login dialog box appears.

**Figure 11–26 ODI Login dialog box - Work Repository**



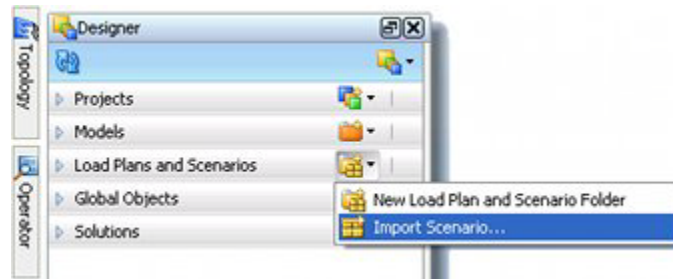
13. Click OK to complete creating the Login.

## Importing the Scenarios

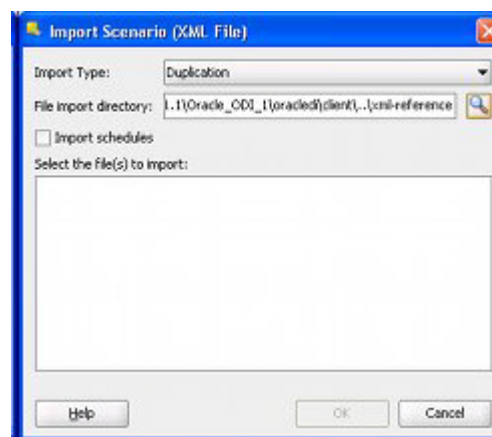
Scenarios contain information on how to process data during the import process.

1. Ensure that the Work Repository is the repository currently connected to.
2. Select the **Designer** navigator.
3. From the **Load Plans and Scenarios** option, select **Import Scenario**.

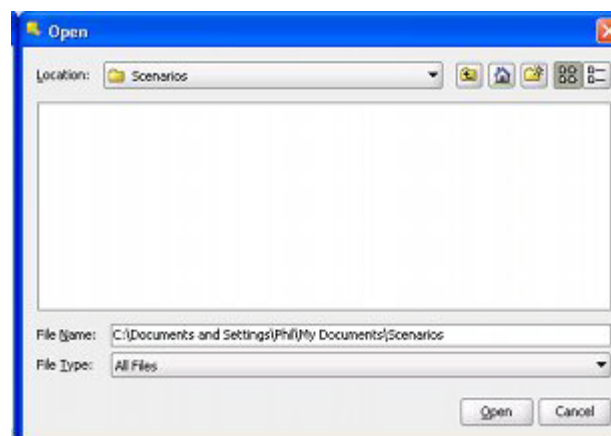


**Figure 11–27 Designer Tab - Import Scenario Option**

4. The Import Scenario dialog box opens.

**Figure 11–28 Import Scenario dialog box - As Opened**

5. Populate the Import Scenario dialog box as follows:
  - a. Set the Import type to **Duplication**.
  - b. Click the magnifying glass to the right of the File Import Directory text box. The **Open** dialog box opens.

**Figure 11–29 Open dialog box**

- c. Use the Location drop down list to navigate to where the scenarios are stored.

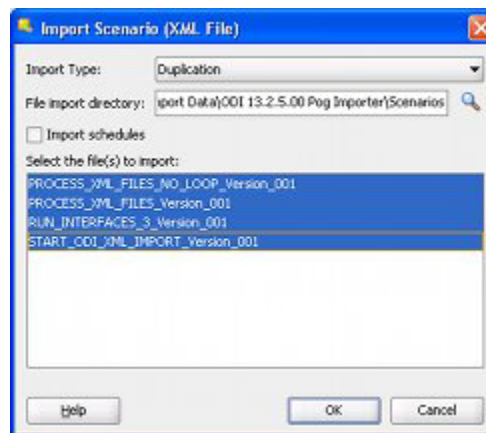
---

**Note:** The Scenarios will not display in the Open dialog box

---

- d. Click **Open**. The Open dialog box will close and the user will be returned to the Import Scenario dialog box with the OK button grayed out.

**Figure 11–30 Import Scenario dialog box - Populated**



- e. Highlight the required scenarios. The OK button will become active. Click **OK**.
- f. On clicking OK, the scenarios will be imported into ODI. An Import Report will be displayed.

Figure 11-31 Import Report.

**Import Report**

Duplication

Imported Objects

Object Type	Object Name	Imported As	Original ID	New ID After Import	Parent Type After Import	Parent Name After Import
Scenario	PROCESS_XML_FILES_NO_LOOP 001	PROCESS_XML_FILES_NO_LOOP 001	289001	1203		
Scenario	PROCESS_XML_FILES 001	PROCESS_XML_FILES 001	290001	2203		
Scenario	RUN_IMPORT 001	RUN_IMPORT 001	278001	3203		
Scenario	RUN_INTERFACES_3 001	RUN_INTERFACES_3 001	76987	4203		
Scenario	START_ODE_XML_IMPORT 001	START_ODE_XML_IMPORT 001	275001	5203		

Deleted Objects

Object Type	Object Name	Original ID
-------------	-------------	-------------

Not Imported Objects

Object Type	Object Name	Original ID
-------------	-------------	-------------

Created Missing References

Description	Object Type	Object Name	Original ID
-------------	-------------	-------------	-------------

Fixed Missing References

Description	Object Type	Object Name	Original ID
-------------	-------------	-------------	-------------

Help Save... Close

- g. Confirm the scenarios have imported satisfactorily and click **Close**.

## Reinstalling ODI and Connecting to Existing Repositories

If ODI Studio is deleted and reinstalled, or if ODI Studio is being installed on another computer, it is not necessary to recreate the Master and Work Repositories. Instead, the connections to the repositories need to be reconfigured.

See the following section for information on how to do this: [Reinstalling ODI and Connecting to Existing Repositories](#).



---

---

## Configuring ODI Agents

This section of the Oracle Data Integrator (ODI) Planogram Import Documentation describes how to configure an ODI Agent. There are two forms of ODI Agent:

- Installed on the same machine as an instance of ODI Studio (local agent).
- Installed by itself on a server (standalone agent).

Agents can be used to execute scheduled imports. Because of latency issues, using the agent installed on the same server as the database instance containing the Macro Space Planning (MSP) database will probably result in the fastest import of data.

The principles of setting up an agent are the same whether it is a local agent or a standalone agent. Either can then be used to execute a scheduled import.

### Installation Permissions

Configuring the ODI agent requires:

- Administrators rights to the machine the agent is installed on.
- A knowledge of where specific files have been installed.
- A knowledge of specific user names and passwords.

### Deployment Details

This section on implementation describes the fundamentals of how to configure the ODI agents with default database schema locations, tables spaces, etc. It also uses the default passwords for ODI.

It is the responsibility of the implementers to modify these details as required in order to obtain the required performance. It is also the responsibility of the implementers to ensure the installation complies with the security policies for the pertinent organizations.

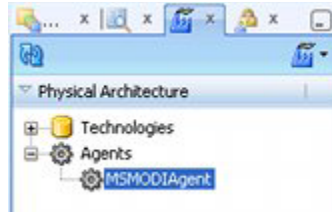
### MSMODI Agent

This agent was imported into ODI Studio when the Master Repository was created. This should be deleted from ODI Studio to avoid confusion with the Standalone Agent created when ODI Standalone Agent was installed. It can be deleted as follows in ODI Studio:

1. Select the **Topology** Navigator.
2. Connect to the **Master Repository**.

3. Open the **Physical Architecture** accordion.
4. Highlight the **MSMODIAgent**. Right click and select **Delete**. The agent will be deleted.

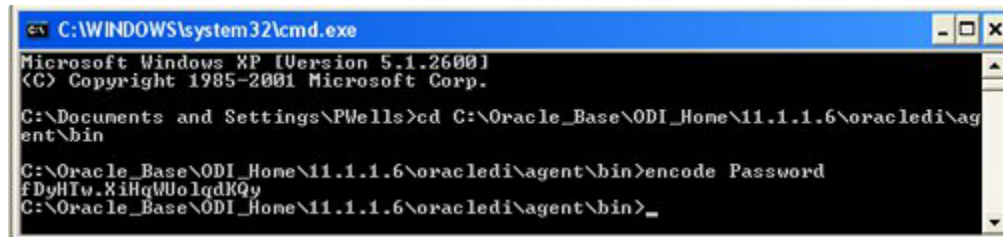
**Figure 12–1 Physical Architecture Accordion - MSMODIAgent**



## Encoding the Passwords

The ExtXML.Properties and **odiparams.bat** files (discussed below) requires an encrypted version of passwords. These passwords can be generated by using the Encode.bat file found in the C:\Oracle\_Base\Oracle\_Home\oracledi\agent\bin directory. This batch file can be found in any installed instance of an ODI agent.

**Figure 12–2 Encoding Passwords**



To encode the password:

1. Open a Command Line prompt
2. Change to the C:\Oracle\_Base\Oracle\_Home\oracledi\agent\bin directory.
3. Use the **encode <password>** command. In the above example, the word **Password** is being encoded.
4. Paste the enciphered password into the pertinent lines of the **odiparamms.bat** batch file.

---

**Note:** The encode.bat file will generate alternative versions of the encrypted password each time it is run.

---

## Configuring Access to the ODI Staging Area

### Purpose

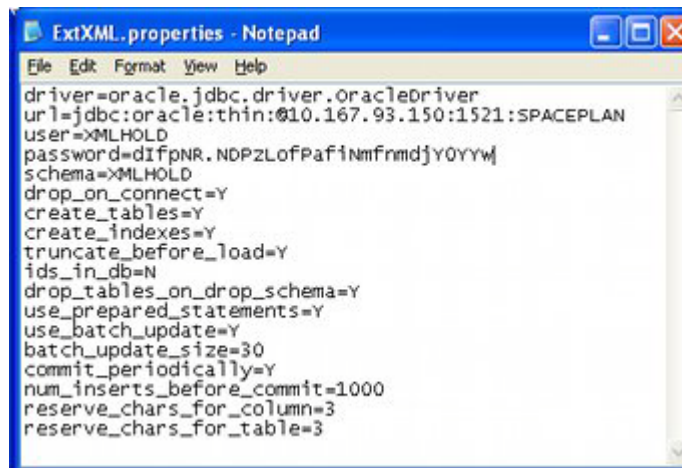
The ODI scenarios for planogram import use the ODI Staging Area to temporarily hold information while executing the import. ODI references the **ExtXML.properties** file for information on how to access the staging area.

### Configuring and Placing the File

The ODI Staging Area is a schema where data is held temporarily while it is validated prior to import in the MSP database. Data for the connection is held in a file called `ExtXML.properties`. For ODI Studio, a copy of the file must exist in `C:\Oracle_Base\Oracle_Home\11.1.1\Oracle_ODI_1\oracledi\agent\drivers` directory.

A copy of the `ExtXML.properties` file is supplied with the installation package. It can be found in the External DB Properties folder.

**Figure 12-3** *ExtXML.Properties File*



Some fields in the supplied file should be updated. This is done as follows:

1. Change the URL to that of the server holding the database instance with the Staging Area schema. This is of the form:
 

```
jdbc:oracle:thin:@<host>:<port>:<sid>
```

  - Host is the IP Address or Computer Name the database instance is hosted on.
  - Port is the port the database listener is configured for. The default is **1521**.
  - SID is the Oracle SID for the database. In this example it is **SPACEPLAN**.
2. Set the User to the user who owns the schema. In this example this is **XMLHOLD**.
3. Set the Schema name to that of the schema. In this example it is **XMLHOLD**.
4. Set the Password to the encoded form for that of the schema. In this example it is **Password\_XMLHOLD**. (See above for how to do this).
5. Save the modified file.
6. Place a copy of the modified file in the required directory.

### Local and Standalone Agents

If both local and standalone agents are installed they will reference the `C:\Oracle_Base\Oracle_Home\11.1.1\Oracle_ODI_1\oracledi\agent\drivers` directory in the location at which they are invoked from.

## Encoding the Password

The ExtXML.properties file requires an encrypted version of the password for the staging area. This can be generated by using the Encode.bat file found in the C:\Oracle\_Base\Oracle\_Home\11.1.1\Oracle\_ODI\_1\oracledi\agent\bin directory for the ODI Standalone Agent.

---

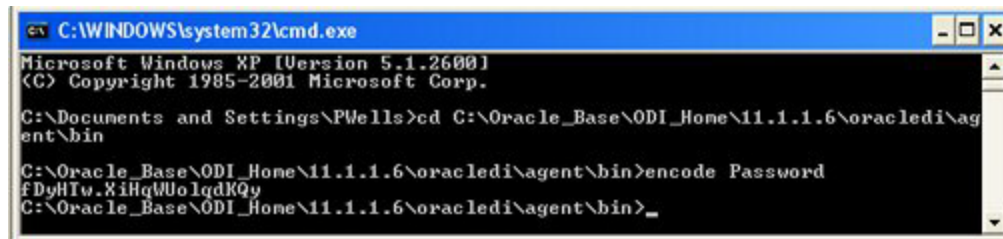
---

**Note:** A copy of the ODI Standalone Agent must be installed to access the **Encode.bat** file. This file is not installed with ODI Studio.

---

---

Figure 12-4 Command Line - Encode Password



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\PWells>cd C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin

C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>encode Password
fDyHTw.XiHqWUo1qdKQy
C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>
```

To encode the password:

1. Open a Command Line prompt.
2. Change to the C:\Oracle\_Base\Oracle\_Home\11.1.1\Oracle\_ODI\_1\oracledi\agent\bin directory.
3. Use the **encode <password>** command. In the above example the word **Password** is being encoded.
4. Paste the enciphered password into the **ExtXML.properties** file.
5. Save the modified file.

---

---

**Note:** The encode.bat file will generate alternative versions of the encrypted password each time it is run.

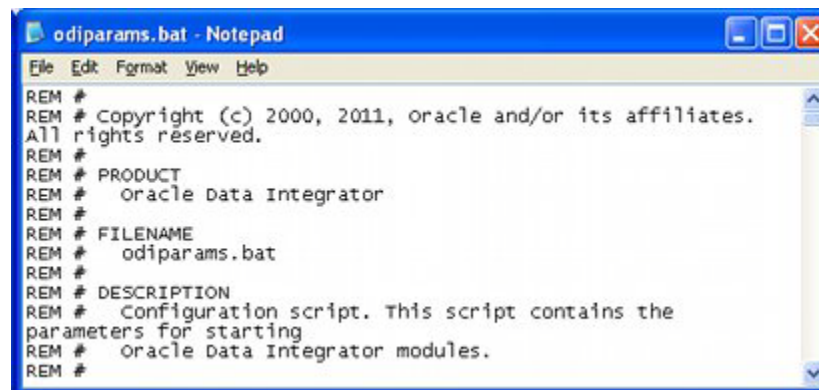
---

---

## Configuring the odiparams Batch File

The **odimarams.bat** file contains the necessary information for the Standalone agent to connect to the Master and Work Repositories. It also contains parameters that allow the Standalone Agent to execute tasks. Finally, the file contains parameters to optimize performance. The odiparams.bat file is located in the C:\Oracle\_Base\Oracle\_Home\oracledi\agent\bin directory on the computer the Standalone agent was installed on.



**Figure 12–5 Odiparams Batch File**


```

odiparams.bat - Notepad
File Edit Format View Help
REM #
REM # Copyright (c) 2000, 2011, oracle and/or its affiliates.
All rights reserved.
REM #
REM # PRODUCT
REM # Oracle Data Integrator
REM #
REM # FILENAME
REM # odiparams.bat
REM #
REM # DESCRIPTION
REM # Configuration script. This script contains the
parameters for starting
REM # Oracle Data Integrator modules.
REM #

```

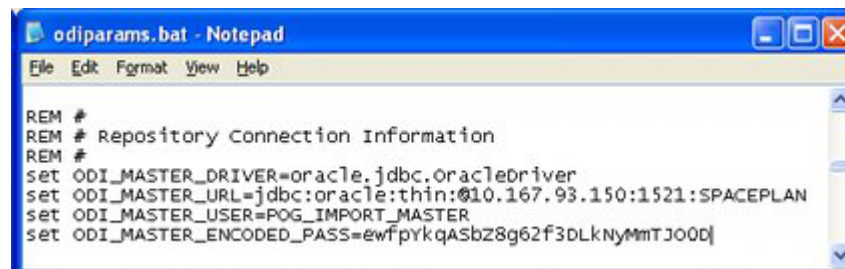
---

**Note:** For more information on the odiparams batch file see the ODI documentation.

---

## Mandatory Parameters

The following parameters must be configured in the odiparams.bat file. These four parameters are required for batch file operations requiring a repository connection (agent, agentstop, startscen, agentscheduler, etc).

**Figure 12–6 Odiparams File - Repository Connection Information**


```

odiparams.bat - Notepad
File Edit Format View Help
REM #
REM # Repository Connection Information
REM #
set ODI_MASTER_DRIVER=oracle.jdbc.OracleDriver
set ODI_MASTER_URL=jdbc:oracle:thin:@10.167.93.150:1521:SPACEPLAN
set ODI_MASTER_USER=POG_IMPORT_MASTER
set ODI_MASTER_ENCODED_PASS=ewfpykqASbZ8g62f3DLkNyMmTJ00Dj

```

### ODI\_MASTER\_DRIVER

This is the JDBC driver used to connect the Master Repository. By default the driver is set to that appropriate to an Oracle database. This is **oracle.jdbc.OracleDriver**.

### ODI\_MASTER\_URL

This is the JDBC URL used to connect the Master Repository. It is in the form:

jdbc:oracle:thin:@<host>:<port>:<sid>

- Host is the IP Address or Computer Name the database instance is hosted on.
- Port is the port the database listener is configured for. The default is 1521.
- SID is the Oracle SID for the database. In this example it is **SPACEPLAN**.

Depending on the precise configuration of the domain the database instance holding the Master Repository Schema is on, it may be necessary to substitute an IP Address for the computer name and remove the internet domain extension from the name of the schema.

---

---

**WARNING:** If an IP Address is substituted for a Computer Name, consult your system administrator to ensure the computer hosting the database instance has a static IP address.

---

---

### **ODI\_MASTER\_USER**

This is the user for the Master Repository. In this example it is **POG\_IMPORT\_MASTER**.

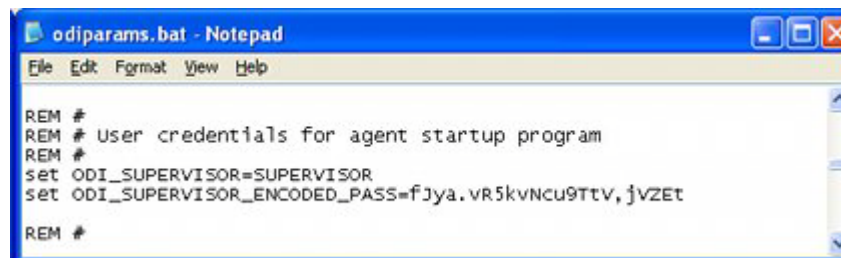
### **ODI\_MASTER\_ENCODED\_PASS**

This is the password for the schema holding the Master Repository. It is encoded by using the **Encode.bat** file in the **C:\Oracle\_Base\Oracle Home\oracledi\agent\bin** directory. In this example it is **Password\_Master**.

## **Parameters for Agent Startup Program**

The two parameters below supply user credentials used only by the agent startup program:

*Figure 12-7 Odiparams File - Agent Startup Information*



```
odiparams.bat - Notepad
File Edit Format View Help
REM #
REM # User credentials for agent startup program
REM #
set ODI_SUPERVISOR=SUPERVISOR
set ODI_SUPERVISOR_ENCODED_PASS=fJya.vR5kVNCu9Ttv, jvZET
REM #
```

### **ODI\_SUPERVISOR**

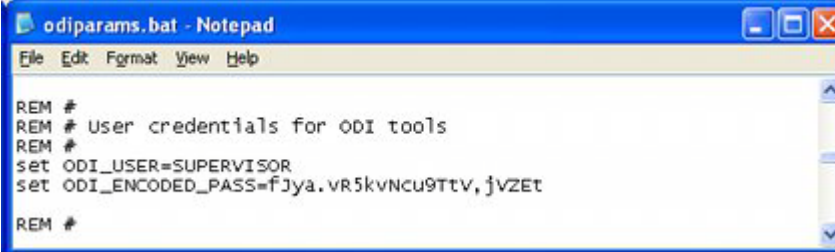
This is the ODI supervisor user name used to gain access to ODI Studio. In this example it is **SUPERVISOR**.

### **ODI\_SUPERVISOR\_ENCODED\_PASS**

This is the password for the SUPERVISOR login for ODI. It is encoded by using the **Encode.bat** file in the **C:\Oracle\_Base\Oracle Home\oracledi\agent\bin** directory. In this example it is **SUNOPSIS**.

## **Parameters for ODI Tool User Credentials**

The two parameters below are used to supply user credentials for all the tools: **startscen**, **restartsession**, **agentstop**, **startcmd**, etc. Unless additional users have been configured in ODI Studio's Security Navigator, the Supervisor logon and its associated password can be used.

**Figure 12–8 Odiparams File - Tool User Credential Information**


```

odiparams.bat - Notepad
File Edit Format View Help
REM #
REM # User credentials for ODI tools
REM #
set ODI_USER=SUPERVISOR
set ODI_ENCODED_PASS=fJya.vR5kvNcu9TtV,jvZET
REM #

```

**ODI\_USER**

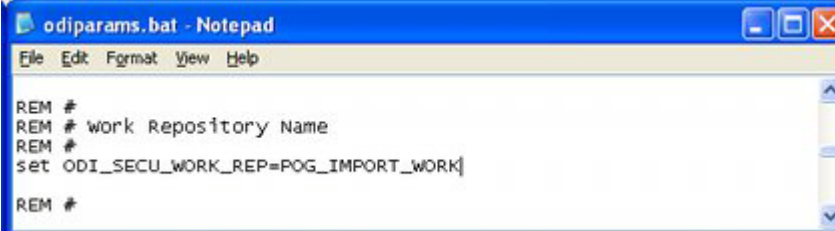
This is the ODI user used to launch a scenario. In this example it is **SUPERVISOR**.

**ODI\_ENCODED\_PASS**

This is the password for the SUPERVISOR login for ODI. It can be encoded by using the **Encode.bat** file in the **C:\Oracle\_Base\Oracle Home\oracledi\agent\bin** directory. In this example it is **SUNOPSIS**.

**Parameter Required by Startcmd, Startscen and Restartsession**

The parameter below is required by startcmd (when the specified command needs to connect to the work repository), startscen, and restartsession scripts. The agent startup and agentstop programs do not need this variable.

**Figure 12–9 Odiparams File - Work Repository Information**


```

odiparams.bat - Notepad
File Edit Format View Help
REM #
REM # work Repository Name
REM #
set ODI_SECU_WORK_REP=POG_IMPORT_WORK
REM #

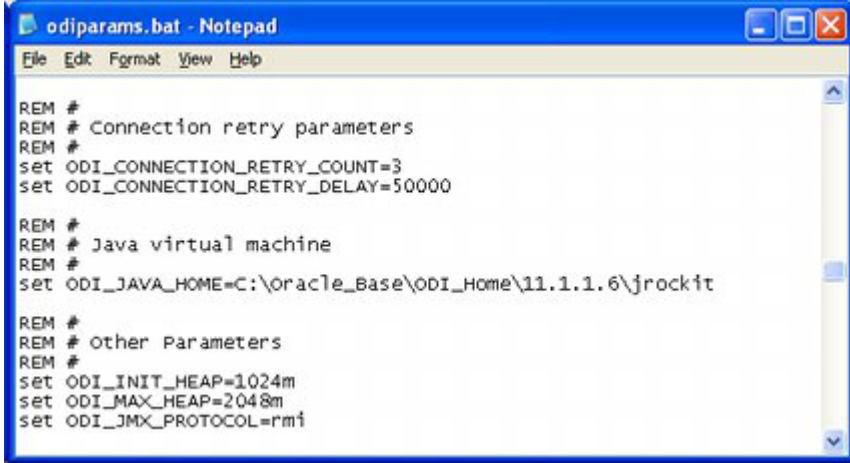
```

**ODI\_SECU\_WORK\_REP**

This is the Name of the Work Repository to connect to. This Work Repository must be attached to the master repository. In this example it is **POG\_IMPORT\_MASTER**.

**Parameters Required to Tune ODI Configuration**

Use the following parameter to tune ODI configuration:

**Figure 12–10 Odiparams File - Tuning Parameters**

```
odiparams.bat - Notepad
File Edit Format View Help
REM #
REM # Connection retry parameters
REM #
set ODI_CONNECTION_RETRY_COUNT=3
set ODI_CONNECTION_RETRY_DELAY=50000

REM #
REM # Java virtual machine
REM #
set ODI_JAVA_HOME=C:\Oracle_Base\ODI_Home\11.1.1.6\jrockit

REM #
REM # Other Parameters
REM #
set ODI_INIT_HEAP=1024m
set ODI_MAX_HEAP=2048m
set ODI_JMX_PROTOCOL=rmi
```

**ODI\_INIT\_HEAP**

This is the minimum size allocated to the Java Virtual Machine (JVM). If this value is set too low, there may be insufficient memory allocated to process the XML file in a reasonable time frame. The default setting is 32m. In this example it has been increased to 1024m.

**ODI\_MAX\_HEAP**

This is the maximum size allocated to the JVM. If this value is set too low, there may be insufficient capacity to process large XML files. The default setting is 256m. In this example it has been increased to 2048m.

---

---

**Note:** The optimum settings for the JVM will vary from machine to machine. Consult your own IT department for case by case values.

---

---

**ODI\_CONNECTION\_RETRY\_COUNT**

This is the number of retries to re-establish master or work repository connections during connection failures. If set to 0, no retry will be performed. In this example it has been set to 3.

**ODI\_CONNECTION\_RETRY\_DELAY**

This is the time in milliseconds between the retries of master or work repository connections. In this example it has been set to 50,000 (5 seconds).

---

---

**Note:** Connection parameters will vary depending on how the ODI Planogram Import solution has been deployed. If the Standalone Agent has been deployed on the same server as the database instance there will probably be few connection problems.

If the Standalone Agent has been deployed on a LAN with poor connectivity to the database server, the parameters may need to be significantly increased.

---

---

### Additional ODI Configuration Parameters

You can use the following parameters to tune performance: `ODI_JAVA_HOME`, `ODI_ADDITIONAL_JAVA_OPTIONS`, `ODI_HOME`, `ODI_JMX_PROTOCOL`. These parameters should only be modified by someone with a good knowledge of Java.

## Configuring the Physical and Logical Agents in ODI Studio

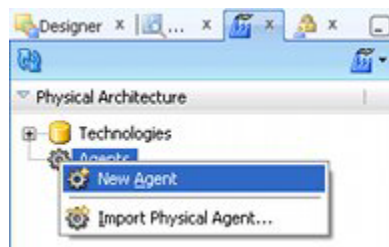
This section specifies how to configure the Standalone agent in ODI Studio.

### Creating the Physical Agent

To create a Physical Agent, do the following:

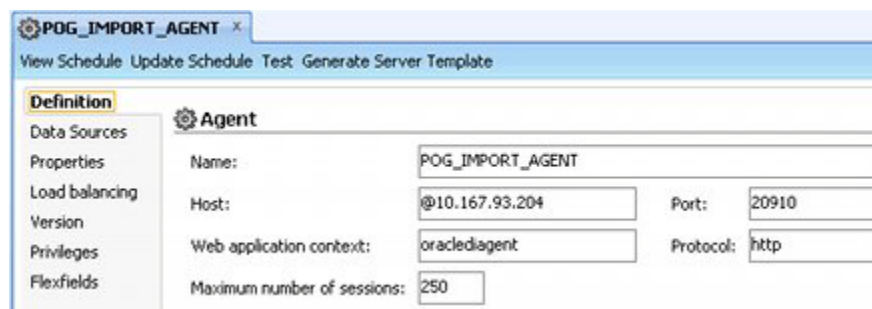
1. Open the **Topology Navigator**.
2. Connect to the **Master Repository** (ODI Menu).
3. Open the **Physical Architecture** accordion by clicking the small triangle to its left.
4. Highlight **Agent**, right click and select **New Agent**.

**Figure 12–11 Topology Navigator - Create New Agent**



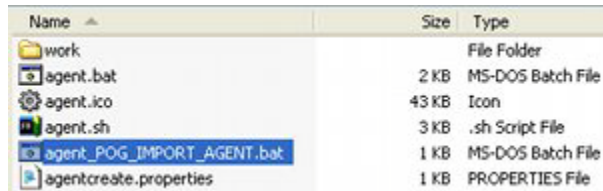
5. The Agent tab will open. Configure it as follows:

**Figure 12–12 Agent Tab**



- a. Set the Name of the agent to that for the agent set up using the Standalone agent. This name must conform to the name of the batch file existing in the `C:\Oracle_Base\Oracle_Home\oracledi\agent\bin` directory where the Standalone Agent was installed.

**Figure 12–13 Standalone Agent - Agent Batch File**



In the above example, the agent has been named **POG\_IMPORT\_AGENT**. This corresponds to the **agent\_POG\_IMPORT\_AGENT.bat** file installed with the standalone agent.

**b.** Set the **Host** as appropriate.

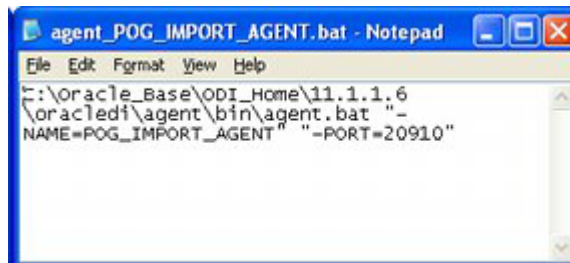
- If the Standalone Agent has been installed on the same computer as ODI Studio, leave the value as **localhost**.

- If the Standalone Agent has been installed on a different computer to ODI Studio, set the host to that remote computer. this can be specified as either a computer name or an IP Address.

In the above example, the host has been set to **@10.167.93.204**.

**c.** Set the Port to the number defined when the Standalone Agent was installed. This value must match that held in the Agent\_<Agent Name>.Bat file.

**Figure 12–14 Agent\_POG\_IMPORT\_AGENT.Bat File**



In this example the port has been set to **20910** in both locations.

**d.** The **Web application context** can be left set to **oraclediagent**.

**e.** The **Protocol** can be left set to **http**.

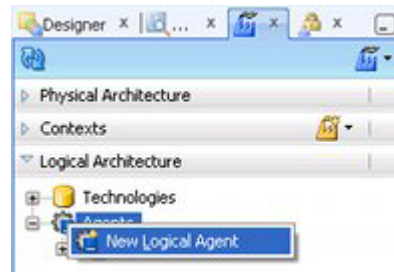
**f.** Set the number of **sessions** to **250**.

**6.** Click **Save** on the ODI Studio toolbar to save these settings. Close the Agent tab.

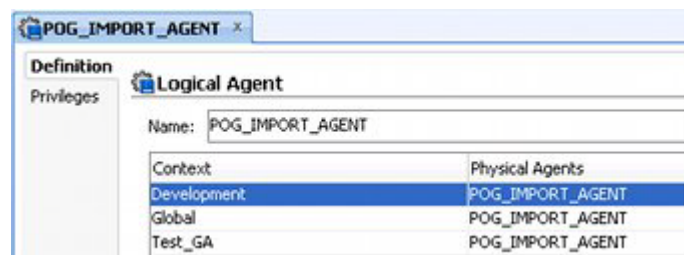
## Creating the Logical Agent

To create the Logical Agent, do the following:

1. Open the **Topology** Navigator.
2. Connect to the **Master Repository** (ODI Menu).
3. Open the **Logical Architecture** accordion by clicking the small triangle to its left.
4. Highlight **Agent**, right click and select **New LogicalAgent**.

**Figure 12–15 Topology Navigator - Create New Logical Agent**

- The Logical Agent tab will open. Configure it as follows:

**Figure 12–16 Logical Agent Tab**

- Set the **Name** of the Logical Agent to match that defined for the Physical Agent.
  - Set the **Physical Agent** for the required Contexts by means of the drop down list. In this example the physical Agents have been set to **POG\_IMPORT\_AGENT**.
- Click **Save** on the ODI Studio toolbar to save these settings. Close the LogicalAgent tab.

## Starting and Testing the Agent

This section deals with starting and testing the Standalone Agent. The procedure is as follows:

- Start the ODI Agent.
- Open ODI Studio, connect to the Master Repository and test the connection to the Standalone Agent in the Topology Navigator.

### Starting the Agent

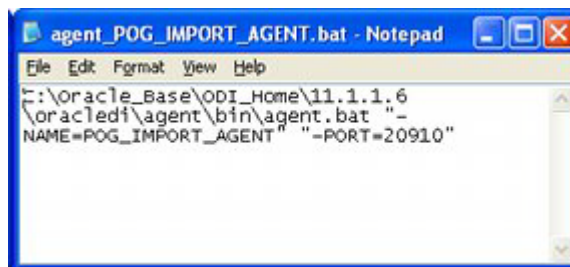
Start the agent as follows:

- In order to run the agent, Open the command prompt and open the directory path where the Agent is present; for example: **C:\Oracle\_Base\Oracle\_Home\oracledi\agent\bin.**

This directory will be located in on the computer the Agent was installed on.

- This directory contains a batch (.BAT) file that invokes the Agent. It will be named after the Agent created in an earlier step. In this example it is **agent\_PLANO\_IMPORT\_AGENT.bat.**




**Figure 12–17 Standalone Agent Batch File**


```
agent_POG_IMPORT_AGENT.bat - Notepad
File Edit Format View Help
C:\Oracle_Base\ODI_Home\11.1.1.6
\oracledi\agent\bin\agent.bat "-
NAME=POG_IMPORT_AGENT" "-PORT=20910"
```

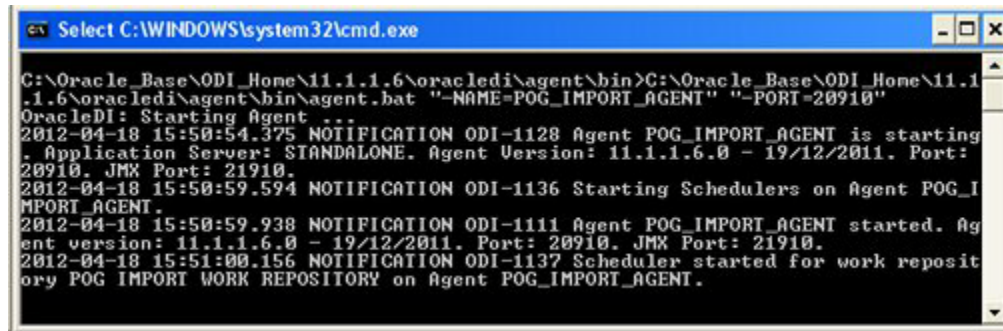
Start the agent from the command line. As the agent starts, it will reference the **odiparams.bat** file for connection information.

3. As the agent is started, a DOS command line windows will appear.

**Figure 12–18 Command Line - Starting Standalone Agent**


```
C:\WINDOWS\system32\cmd.exe
C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>C:\Oracle_Base\ODI_Home\11.1
.1.6\oracledi\agent\bin\agent.bat "-NAME=POG_IMPORT_AGENT" "-PORT=20910"
OracleDI: Starting Agent ...
```

After a brief pause, the agent will start; the required confirmation appearing in the DOS Command Line window.

**Figure 12–19 Command Line - Standalone Agent Started**


```
Select C:\WINDOWS\system32\cmd.exe
C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>C:\Oracle_Base\ODI_Home\11.1
.1.6\oracledi\agent\bin\agent.bat "-NAME=POG_IMPORT_AGENT" "-PORT=20910"
OracleDI: Starting Agent ...
2012-04-18 15:50:54.375 NOTIFICATION ODI-1128 Agent POG_IMPORT_AGENT is starting
. Application Server: STANDALONE. Agent Version: 11.1.1.6.0 - 19/12/2011. Port:
20910. JMX Port: 21910.
2012-04-18 15:50:59.594 NOTIFICATION ODI-1136 Starting Schedulers on Agent POG_I
MPORT_AGENT.
2012-04-18 15:50:59.938 NOTIFICATION ODI-1111 Agent POG_IMPORT_AGENT started. Ag
ent version: 11.1.1.6.0 - 19/12/2011. Port: 20910. JMX Port: 21910.
2012-04-18 15:51:00.156 NOTIFICATION ODI-1137 Scheduler started for work reposit
ory POG_IMPORT_WORK_REPOSITORY on Agent POG_IMPORT_AGENT.
```

4. Leave the Standalone Agent running while the connection is tested from ODI Studio.

## Testing the Agent from ODI Studio

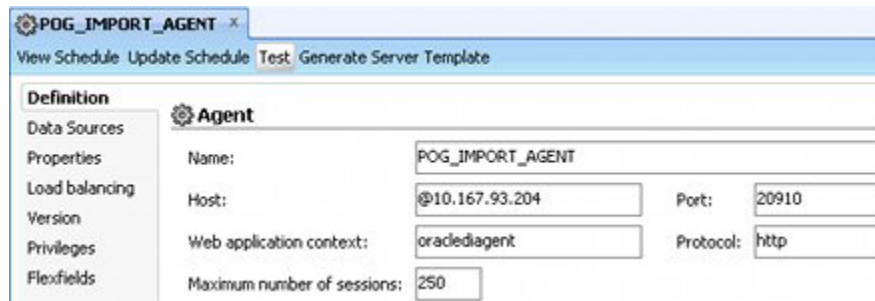
The Standalone agent can be tested in ODI Studio as follows:

1. Open the Topology Navigator and connect to the Master Repository.
2. In the Physical Architecture accordion right click on the pertinent Agent and select Open for more details.

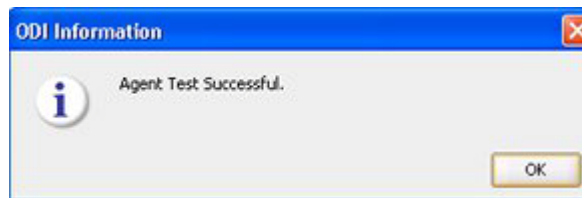


**Figure 12–20 Topology Navigator - Available Agents**

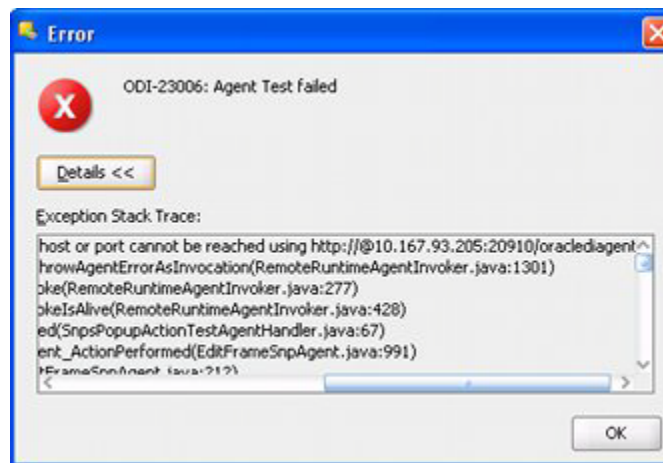
3. The Agent's dialog box will appear. Select the **Definition** Tab.

**Figure 12–21 Physical Agent - Definition Tab**

4. Click **Test**. The connection to the Standalone Agent will now be tested. A dialog box will report the results.
  - If the connection is successful, it will be reported in a dialog box.

**Figure 12–22 Standalone Agent Connection Confirmation**

- If the connection fails, an error message will result.

**Figure 12–23 Standalone Agent Connection Error Dialog Box****Sources of Connection Errors**

Sources of potential connection errors are:

- The Standalone Agent has not started correctly.
- The connection details in the Definition tab of the Physical Agent in ODI Studio are incorrect.
- One or more parameters in the odiparams.bat file are incorrect.

**ODI and Scheduling**

Once the planogram import process is ready to go live, it will probably be set up to execute on a regular schedule. This can be done in three main ways:

- Using ODI Studio's (or ODI Console's) internal scheduling tool
- ODI Agent
- ODI Standalone Agent

If the Agent or Standalone Agent is used, it can be initiated in one of two ways:

- By means of a batch file
- By starting the appropriate Windows Service

This is discussed in more detail in the section on running planogram imports: [Running ODI Planogram Imports](#).

---

---

## Running ODI Planogram Imports

This section describes how to run planogram imports. It describes three different methods of executing those imports.

### Overview

This section gives an overview of Planogram Imports.

### Forms of Planogram Import

There are two basic forms of planogram import.

#### **Bulk Loading**

Bulk loading of planograms occurs when it is necessary to import the entire list of planograms into the Macro Space Planning (MSP) database. A typical instance for this would be when planograms are being loaded for the first time into the MSP database. This may take an appreciable time.

#### **Loading Updated Planograms**

Updating of planograms occurs at regular intervals to bring in new and updated planograms. Data volumes will be lower than for a bulk loading exercise.

### Running Planogram Imports

There are several ways planogram imports can be run.

#### **Directly from ODI Studio**

Planogram imports can be run manually from the Operator Navigator within ODI Studio.

#### **Run to a Schedule using ODI Studio and an Agent**

Planogram imports can be scheduled at specific times using the scheduling tools within ODI studio and a local or standalone agent.

#### **Run via the Command Line**

Planogram imports can be run from the command line using a local or standalone agent and a scheduling tool.

## Manually Running a Planogram Import from ODI Studio

This section explains how to execute a planogram import manually from ODI Studio.

### Preparatory Actions

The following must have been carried out prior to running a planogram import:

- The specified import and archive directories must be local to ODI Studio.
- The system variables describing these directories (MSP Administration Module) are pointing to the correct directories.
- A copy of the XML file containing the planogram details must exist in the specified import directory.
- The appropriate scenarios must be loaded into the pertinent Work Repository.

---



---

**Note:** The Import and Archive directories need to be local to the initiating application (in this case ODI Studio). If another location has been used for the previous import, it may be necessary to modify the paths held in the system variables describing the directories when the initiating application changes.

---



---

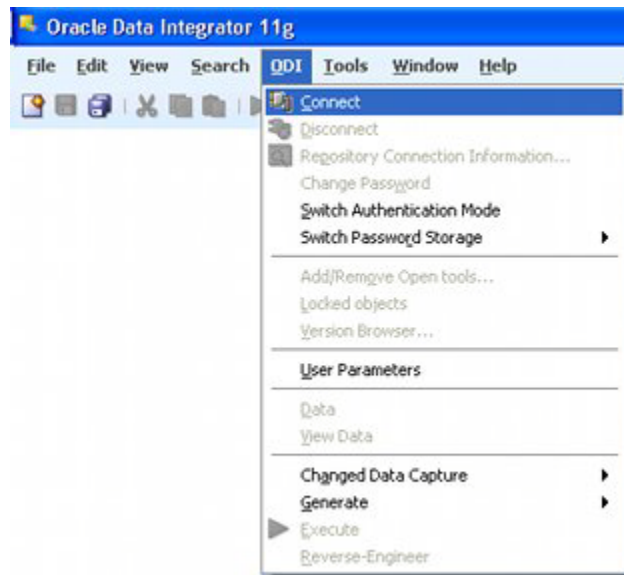
### Running the Planogram Import

These actions describe how to run a manual planogram import.

#### Connect to the Work Repository

The Work Repository holds the details of how to execute the import. This can be connected within ODI Studio by selecting the **Connect** option on the ODI menu.

*Figure 13–1 ODI Menu - Connecting to Work Repository*



---

---

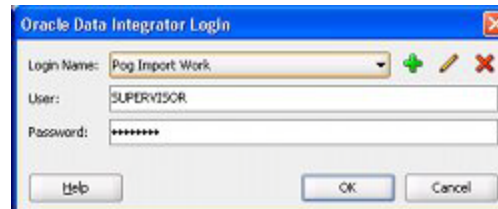
**Note:** If ODI is currently connected to another repository, this repository can be disconnected through the **Disconnect** option on the ODI menu.

---

---

The ODI Login dialog box will appear.

**Figure 13–2 ODI Login Dialog Box**



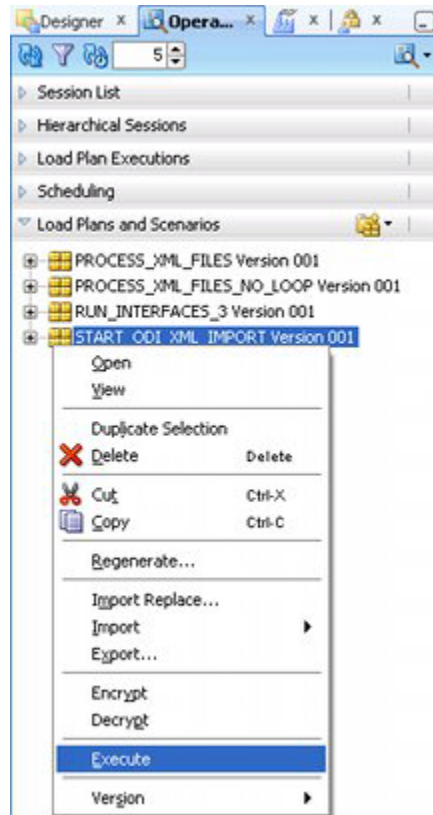
Select the required Work Repository from the **Login Name** drop down list. Click OK and a connection will be made to the required Work Repository.

### Starting a Scenario from the Operator Navigator

Having connected to the Work Repository, the data import is executed from the **Operator Navigator**. It is carried out as follows:

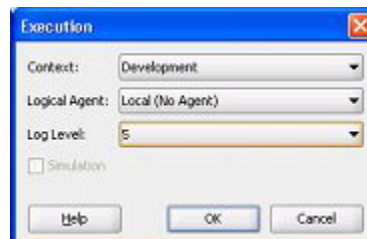
1. Open the **Load Plans and Scenarios** folder by clicking the small triangle to the left of the title. A list of scenarios will be displayed. Right click on the **START\_ODI\_XML\_IMPORT** scenario and select the **Execute** option.

**Figure 13–3 Load Plans and Scenarios Folder - Execute Option**



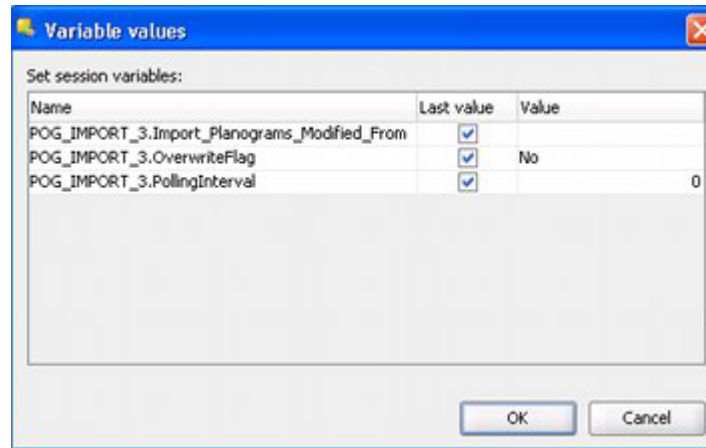
2. The Execution dialog box will appear.

**Figure 13–4 Execution Dialog Box**



Set the drop down lists as follows:

- a. Set the **Context** to the environment the import will be carried out in.
  - b. Set the **Logical Agent** to Local (No Agent).
  - c. Set the **Log Level** to the required level of detail
- Click **OK**.
3. The Variable Values dialog box will appear.

**Figure 13–5 Variable Values Dialog Box**

There are two ways of using this dialog box:

- **Last Value:** if this check box is ticked, the value used in the previous import will be read from the Work Repository and used.
- **Value:** If the Last Value check box is not checked, the data in the value column will be used for the planogram import.

Data for the Value field is entered as followed:

- **Import\_Planograms\_Modified\_Since:** if this field is populated, any planograms imported will be checked to see the last modified date exceeds this value. The variable will be in the format MMDDYYYY hh:mm:ss format, where MM = a two digit month, DD = a two digit day, YYYY = a four digit year, hh = a two digit hour, mm = a two digit minute and ss = a two digit second.
- **Overwrite Flag:** this flag can to update existing planogram records. If set to Yes and if the planogram already exists in the MSP database, the existing values will be overwritten with those being imported. If set to No, updating will not take place.

Any new planograms which are not present in the MSP database will be imported irrespective of the Overwrite Flag value.

- **Polling Interval:** this is used to check if fresh files have been added to the Import Directory during the time the import process is running. It is in minutes and can be set to any value from 0 to 60. If set to 0, the Import Directory will only be checked when the Import Process is started.

Click **OK**.

4. The Session Started dialog box will appear. Click **OK** to confirm and start the session.
5. The results of operations ODI carried out during the data import can be seen in the Operator Navigator.

**Figure 13–6 Operator Navigator - Results**



6. The results of importing data values into the MSP database are written to the AVTTB\_ODI\_LOG table.

**Figure 13–7 AVTTB\_ODI\_LOG Table**

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ODL_ID	NUMBER	No	1	1	(null)
ODL_TEXT	VARCHAR2(4000 BYTE)	No	(null)	2	(null)
ODL_START_DATE	DATE	No	(null)	3	(null)
ODL_TYPE	NUMBER	Yes	(null)	4	(null)
ODL_STOP_DATE	DATE	Yes	(null)	5	(null)
ODL_FILENAME	VARCHAR2(4000 BYTE)	Yes	(null)	6	(null)

This holds information such as the error messages written and their date, together with an enumerated value signifying the severity of the error.

## Running using ODIs Scheduling Tool and an Agent

This section describes how planogram imports can be scheduled using ODIs scheduling tool.

### Preparatory Actions

The following must have been carried out prior to running a planogram import:

- The specified import and archive directories must be local to the Agent that will be used to trigger the imports. This agent may local (on the same machine as ODI Studio) or standalone (typically on a server and separate from ODI Studio).
- The system variables describing these directories (MSP Administration Module) are pointing to the correct directories.
- The odiparams file is correctly configured.
- A copy of the XML file containing the planogram details must exist in the specified import directory.
- The appropriate scenarios must be loaded into the pertinent Work Repository.

---

**Note:** The Import and Archive directories need to be local to the initiating application (the Standalone Agent). Thus, it may be necessary to modify the paths held in the system variables describing the directories when the initiating application changes.

---



## Setting up the Schedule

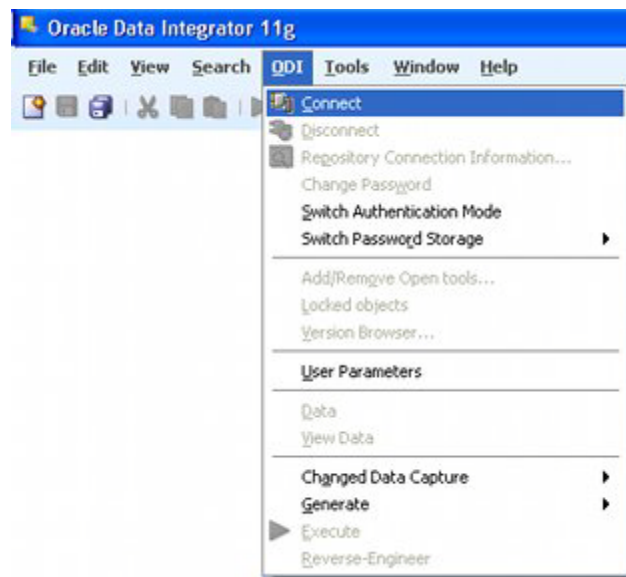
The schedule for the planogram imports is set up within ODI Studio as follows:

1. Connect to the Work Repository.
2. Specify the schedule in the Designer Navigator.
3. Update the Schedule for the Agent.
4. Start the Agent.
5. Verify the Import.

## Connect to the Work Repository in ODI Studio

The Work Repository holds the details of how to execute the import - including the schedule. This can be connected within ODI Studio by selecting the **Connect** option on the ODI menu.

**Figure 13–8** ODI Menu - Connecting to Work Repository



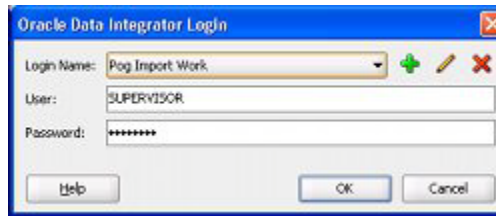
---

**Note:** If ODI is currently connected to another repository, this repository is disconnected via the **Disconnect** option on the ODI menu.

---

The ODI Login dialog box will appear.

**Figure 13–9 ODI Login Dialog Box**



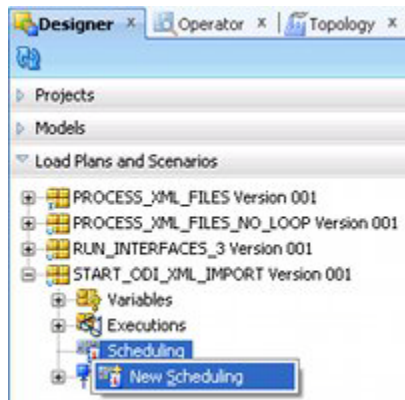
Select the required Work Repository from the **Login Name** drop down list. Click OK and a connection will be made to the required Work Repository.

## Specifying the Schedule in the Designer or Operator Navigator

The schedule is specified as follows:

1. In the **Designer or Operator Navigator** open the **Load Plans and Scenarios** option by clicking the small rectangle to the left of the title. Expand the options for the **Start\_ODI\_XML\_Import** scenario. Highlight **Scheduling**, right click and select **New Scheduling**.

**Figure 13–10 Designer Navigator - Selecting New Schedule**



2. The Scenario Scheduling Tab will open.

Figure 13–11 Scenario Scheduling Tab

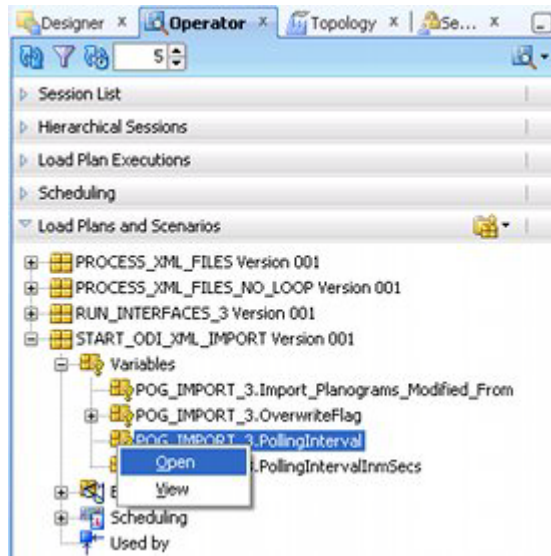
The screenshot shows the 'Scenario Scheduling' configuration window for the scenario 'START\_ODI\_XML\_IMPORT / 001'. The window is divided into three main sections:

- Definition:** Contains dropdown menus for 'Context' (set to 'Development'), 'Agent' (set to 'POG\_IMPORT\_AGENT'), and 'Log Level' (set to '5').
- Status:** Includes radio buttons for 'Active', 'Inactive', and 'Active for the period' (which is selected). Below this are options for scheduling: 'Starting' and 'Ending' (both set to '23-Apr-2012' at '13:25:39'), 'Every day between' (set to '13:25:39' to '13:25:39'), and 'Except these days of the week' (with 'Sunday' selected).
- Execution:** Includes radio buttons for 'On startup', 'Simple', 'Hourly', 'Daily' (selected), 'Weekly', 'Monthly (day of the month)', 'Monthly (week day)', and 'Yearly'. A 'Time' field is set to '00:15:00'.

This is configured as follows:

- a. Use the drop down list to set the **Context** to the environment where the planogram import is going to be carried out. In this example it is the Development environment.
  - b. Set the **Log Level** to that required. In this example 5 levels have been selected from the drop down list - this will result in detailed logging of results.
  - c. Set the **Agent** to the required standalone agent from the drop down list. In this example it is **POG\_IMPORT\_AGENT**.  
This is a standalone agent. It would also be possible to set the agent to a local agent if required.
  - d. Set the **Status** to what is required. There are many options available for this - in the above example, the planogram import has been set to execute every day except Sunday.
  - e. Set the required **Execution** interval. Again, there are many options available. In the above example, the import has been set to execute at 15 minutes past midnight on the specified days.
3. If it is required to set a Polling Interval, this may be done as follows:
    - a. In the Operator Navigator, open **START\_ODI\_XML\_IMPORT** scenario.
    - b. Highlight the **POG\_IMPORT\_\_3.PollingInterval** option, right click and select Open.

**Figure 13–12 Opening Polling Interval Variable**



- c. The Default interval is then set. In this example, checks will be made for new XML files every minute while the import is active.

**Figure 13–13 Setting Polling Interval Variable**



- 4. Click **Save** on the ODI toolbar to save the schedule.

---

**Note:** Saving the schedule has only saved the information. It has not yet made it available to the Agent.

---

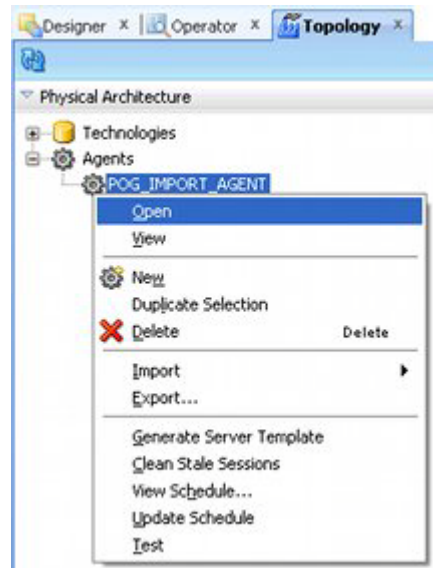
- 5. Start the Agent - see the next section for information on how to do this: [Starting and Stopping the Standalone Agent](#).

---

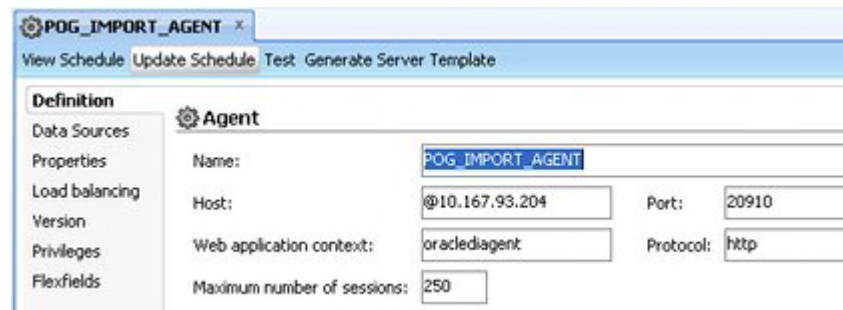
**Note:** This agent may be either local or standalone - depending on where the import is to be initiated from.

---

- 6. In the Topology Navigator, open the Physical Architecture accordion by clicking the small triangle to the left of the name. Open up the list of agents and highlight the required standalone agent. Right click and select Open.

**Figure 13–14 Topology Navigator - Opening Agent to Update**

7. The Agent Tab will appear. Click the **Update Schedule** option from the toolbar.

**Figure 13–15 Agent Tab - Update Schedule Option**

8. The Select Repositories dialog box will appear. Select the Work Repositories the schedule is to be saved to.

**Figure 13–16 Select Repositories dialog box**

On clicking OK, the scheduling information will be updated in the pertinent work repositories and a confirmatory dialog box will appear. This information can now be accessed by the Standalone agent when it is started.

## Starting and Stopping the Standalone Agent

Once the standalone agent has been started, it will link to the specified work repository and execute the required instructions.

---

**Note:** there are varying ways of starting agent through the command line. See the ODI documentation for more information. Also see information on DOS syntax.

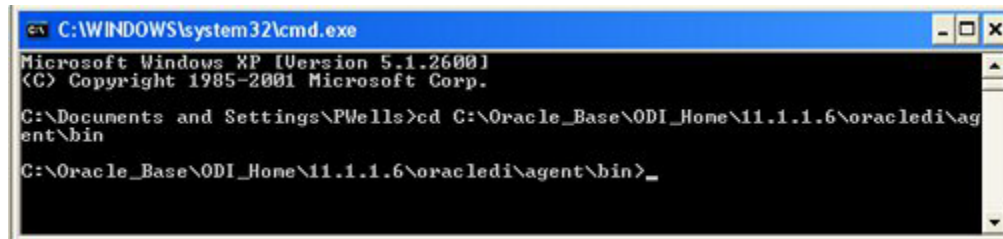
---

### Starting the Standalone Agent

The standalone agent is started as follows on the machine holding it.

1. Open a Command Line window. Change the default directory to the `C:\Oracle_Base\ODI_Home\oracledi\agent\bin` directory holding the batch file for the standalone agent.

*Figure 13–17 Command Line - Changing the Directory*



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

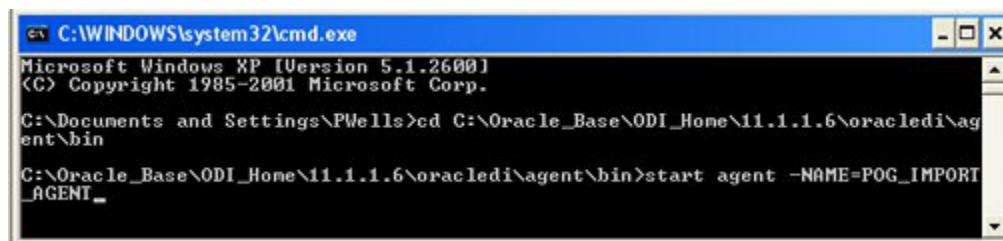
C:\Documents and Settings\PWells>cd C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin

C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>_

```

2. Start the standalone agent with the command `start agent -NAME=POG_IMPORT_AGENT`.

*Figure 13–18 Command Line - Starting the Standalone Agent*



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\PWells>cd C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin

C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>start agent -NAME=POG_IMPORT_AGENT_

```

The syntax for this can be deconstructed as follows:

- **Start** is a command instructing that the standalone agent be started in a new command window.
- **agent** is the batch file that is used to start standalone agents.
- **-NAME=POG\_IMPORT\_MASTER** is a parameter used to start a specific standalone agent. **-NAME** is the name of the parameter: it is preceded by a '-'

sign and should be in capitals. The '=' sign and the information following it is the name of the standalone agent: in this example **POG\_IMPORT\_MASTER**.

3. The Standalone Agent will then start in a new command window.

**Figure 13–19 Command Line - Running Agent**

```

C:\WINDOWS\system32\cmd.exe - agent -NAME=POG_IMPORT_AGENT
OracleDI: Starting Agent ...
2012-04-23 15:16:32.891 NOTIFICATION ODI-1128 Agent POG_IMPORT_AGENT is starting
. Application Server: STANDALONE. Agent Version: 11.1.1.6.0 - 19/12/2011. Port:
20910. JMX Port: 21910.
2012-04-23 15:16:38.031 NOTIFICATION ODI-1136 Starting Schedulers on Agent POG_I
MPORT_AGENT.
2012-04-23 15:16:38.344 NOTIFICATION ODI-1111 Agent POG_IMPORT_AGENT started. Ag
ent version: 11.1.1.6.0 - 19/12/2011. Port: 20910. JMX Port: 21910.
2012-04-23 15:16:38.547 NOTIFICATION ODI-1137 Scheduler started for work reposi
tory POG_IMPORT_WORK_REPOSITORY on Agent POG_IMPORT_AGENT.

```

---

**Note:** The Standalone Agent must be stopped in a separate command window - the **start** command when initiating it keeps the original command window available for this purpose.

---

4. All the time the standalone agent is running, it will execute any scheduled tasks.

#### Other Command Line Options

The following Command Line options are available:

- **Log Level** is the level of logging required. This can be set from 0 - 5.
- **-AGENT\_URL = <Remote\_Agent\_URL>**. This command can be used to start an agent on a remote computer. If this parameter is used the NAME parameter is ignored.
- **-SESSION\_NAME=<Session\_Name>**. This parameter can be used to assign a name to the session in the ODI Session List.
- **-KEYWORDS=<keywords>**. This option can be used if it is required to attach additional information to a session. It takes the form of a comma separated list.

#### Stopping the Standalone Agent

To stop a standalone agent:

1. Open a command line window and change the directory to the **C:\Oracle\_Base\ODI\_Home\oracledi\agent\bin** directory.
2. Stop the standalone agent with the command **agentstop -NAME=POG\_IMPORT\_AGENT**.



**Figure 13–20 Command Line - Stopping Agent - 1**

```
C:\WINDOWS\system32\cmd.exe
C:\Oracle_Base\ODI_Home\11.1.1.6\oracle\odi\agent\bin>agentstop -NAME=POG_IMPORT_AGENT
```

The syntax for this can be deconstructed as follows:

- **agentstop** is the batch file that is used to stop standalone agents.
  - **-NAME=POG\_IMPORT\_MASTER** is a parameter used to start a specific standalone agent. **-NAME** is the name of the parameter: it is preceded by a '-' sign and should be in capitals. The '=' sign and the information following it is the name of the standalone agent: in this example **POG\_IMPORT\_MASTER**.
3. Confirmation is written to the command line window showing the standalone agent has stopped.

**Figure 13–21 Command Line - Stopping Agent - 2**

```
C:\WINDOWS\system32\cmd.exe
C:\Oracle_Base\ODI_Home\11.1.1.6\oracle\odi\agent\bin>agentstop -NAME=POG_IMPORT_AGENT
2012-04-23 15:46:13.766 NOTIFICATION OracleDI: Stopping Agent POG_IMPORT_AGENT..
C:\Oracle_Base\ODI_Home\11.1.1.6\oracle\odi\agent\bin>
```

4. A prompt will also appear in the command line window for the standalone agent confirming the process has stopped.

**Figure 13–22 Command Line - Standalone Agent Stop Confirmation**

```
C:\WINDOWS\system32\cmd.exe
OracleDI: Starting Agent ...
2012-04-23 15:44:41.234 NOTIFICATION ODI-1128 Agent POG_IMPORT_AGENT is starting
. Application Server: STANDALONE. Agent Version: 11.1.1.6.0 - 19/12/2011. Port:
20910. JMX Port: 21910.
2012-04-23 15:44:46.406 NOTIFICATION ODI-1136 Starting Schedulers on Agent POG_I
MPORT_AGENT.
2012-04-23 15:44:46.719 NOTIFICATION ODI-1111 Agent POG_IMPORT_AGENT started. Ag
ent version: 11.1.1.6.0 - 19/12/2011. Port: 20910. JMX Port: 21910.
2012-04-23 15:44:46.859 NOTIFICATION ODI-1137 Scheduler started for work reposit
ory POG_IMPORT_WORK_REPOSITORY on Agent POG_IMPORT_AGENT.
2012-04-23 15:46:17.922 NOTIFICATION ODI-1130 Agent POG_IMPORT_AGENT is stopping
.
2012-04-23 15:46:17.938 NOTIFICATION ODI-1129 Agent POG_IMPORT_AGENT stopped.
C:\Oracle_Base\ODI_Home\11.1.1.6\oracle\odi\agent\bin>
```

5. Once the Standalone Agent has stopped, it will no longer execute scheduled tasks.



## Running Scheduled Tasks using Batch Files

Scheduled tasks can be executed using batch files. The section below shows a basic example. More sophisticated task can be executed by implementers with a knowledge of Windows Batch File syntax.

### Preparatory Actions

The following must have been carried out prior to running a planogram import:

- The specified import and archive directories must be local to the Standalone Agent that will be used to trigger the imports.
- The system variables describing these directories (MSP Administration Module) are pointing to the correct directories.
- The odiparams file is correctly configured.
- A copy of the XML file containing the planogram details must exist in the specified import directory.
- The appropriate scenarios must be loaded into the pertinent Work Repository.

---

---

**Note:** The Import and Archive directories need to be local to the initiating application (the Standalone Agent). Thus, it may be necessary to modify the paths held in the system variables describing the directories when the initiating application changes.

---

---

### Setting up the Schedule

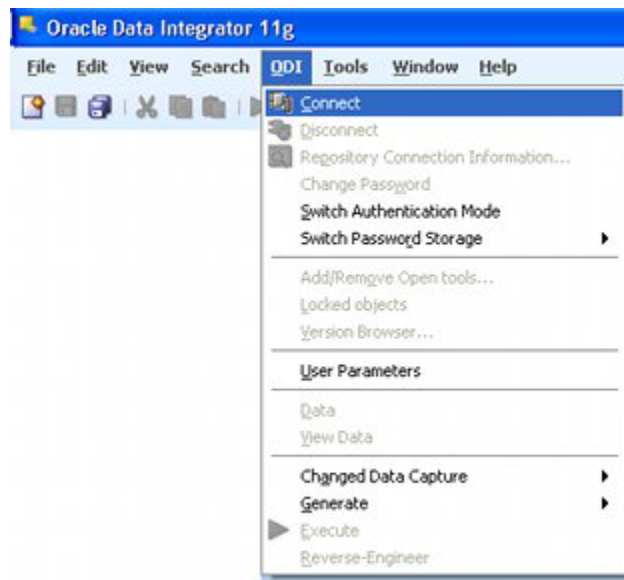
The schedule for the planogram imports is set up within ODI Studio as follows:

1. Connect to the Work Repository
2. Specify the schedule in the Designer Navigator
3. Update the Schedule for the Agent
4. Use a batch file to start the Agent at a specified time
5. Use a batch file to stop the agent at a specified time
6. Verify the Import

### Connect to the Work Repository in ODI Studio

The Work Repository holds the details of how to execute the import - including the schedule. This can be connected within ODI Studio by selecting the **Connect** option on the ODI menu.

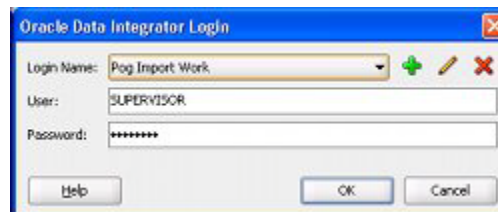
**Figure 13–23 ODI Menu - Connecting to Work Repository**



**Note:** If ODI is currently connected to another repository, this repository can be disconnected via the **Disconnect** option on the ODI menu.

The ODI Login dialog box will appear.

**Figure 13–24 ODI Login Dialog Box**



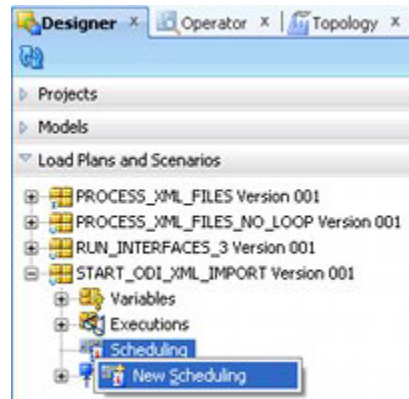
Select the required Work Repository from the **Login Name** drop down list. Click OK and a connection will be made to the required Work Repository.

## Specifying the Schedule in the Designer or Operator Navigator

The schedule is specified as follows:

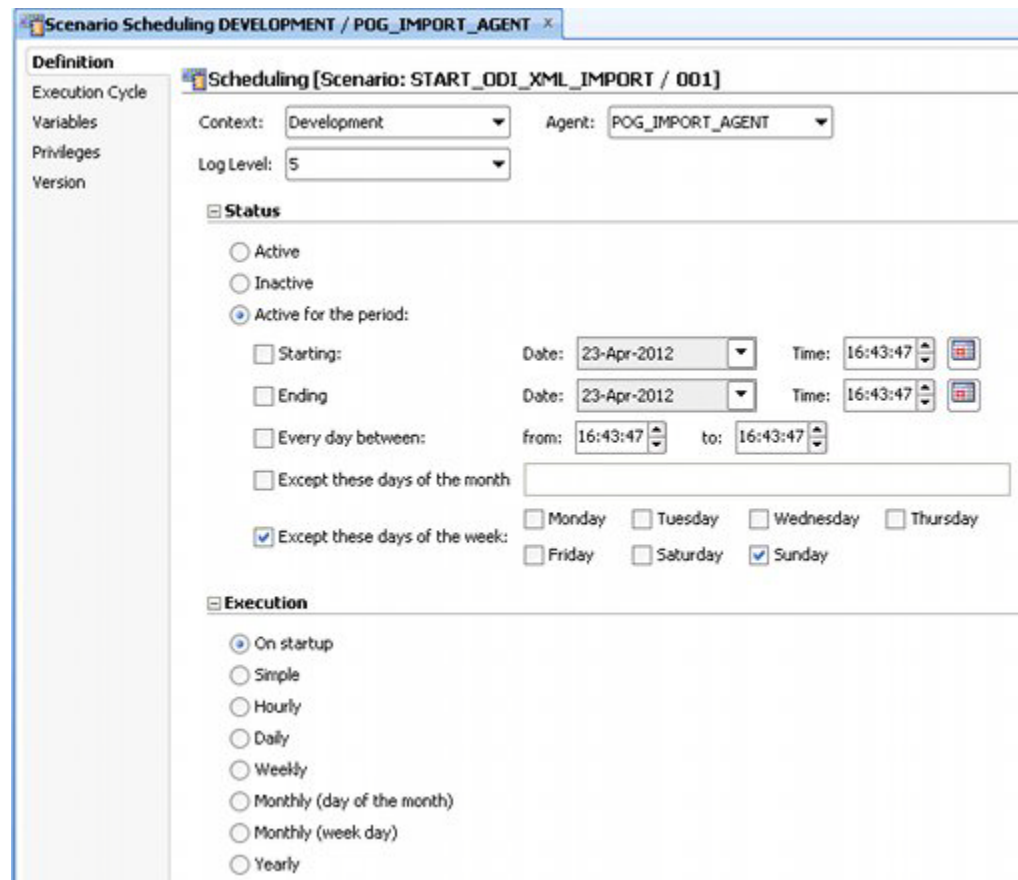
1. In the **Designer or Operator Navigator** open the **Load Plans and Scenarios** option by clicking the small rectangle to the left of the title. Expand the options for the **Start\_ODI\_XML\_Import** scenario. Highlight **Scheduling**, right click and select **New Scheduling**.

Figure 13–25 Designer Navigator - Selecting New Schedule



2. The Scenario Scheduling Tab will open.

Figure 13–26 Scenario Scheduling Tab - Using Batch Files



This is configured as follows:

- a. Use the drop down list to set the **Context** to the environment where the planogram import is going to be carried out. In this example it is the Development environment.

- b. Set the **Log Level** to that required. In this example 5 levels have been selected from the drop down list - this will result in detailed logging of results.
  - c. Set the **Agent** to the required standalone agent from the drop down list. In this example it is POG\_IMPORT\_AGENT.
  - d. Set the **Status** to what is required. There are many options available for this - in the above example, the planogram import has been set to execute every day except Sunday.
  - e. Set the required **Execution** interval. This needs to be set to On Start-Up. this means the Planogram Import will initiate whenever the Standalone Agent is started.
3. Click **Save** on the ODI toolbar to save the schedule.

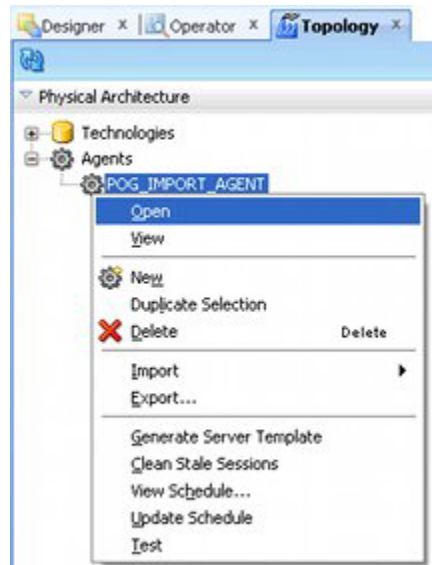
---

**Note:** Saving the schedule has only saved the information. It has not yet made it available to the Standalone Agent.

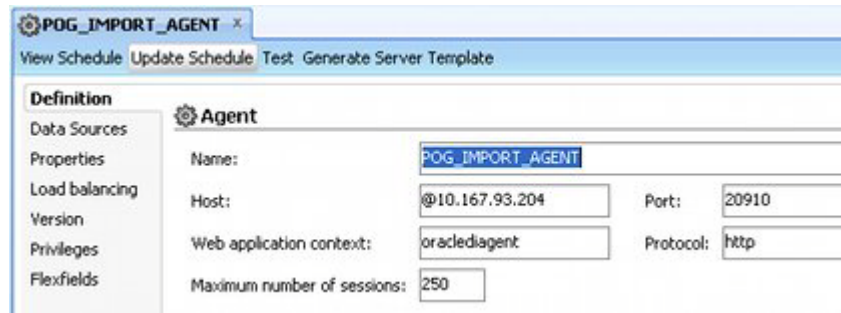
---

- 4. Start the Standalone Agent - see the earlier section for information on how to do this.
- 5. In the Topology Navigator, open the Physical Architecture accordion by clicking the small triangle to the left of the name. Open up the list of agents and highlight the required standalone agent. Right click and select Open.

**Figure 13–27 Topology Navigator - Opening Agent to Update**



- 6. The Agent Tab will appear. Click the **Update Schedule** option from the toolbar.

**Figure 13–28 Agent Tab - Update Schedule Option**

7. The Select Repositories dialog box will appear. Select the Work Repositories the schedule is to be saved to.

**Figure 13–29 Select Repositories dialog box**

On clicking OK, the scheduling information is updated in the pertinent work repositories and a confirmatory dialog box will appear. This information can now be accessed by the Standalone agent when it is started.

## Controlling the Import Schedule using Batch Files

Once the schedule has been set to **On startup**, the planogram import is controlled by stopping and starting the agent. this in turn can be done by invoking batch files at specific times by means of an external scheduling tool.

---

**Note:** This example uses very simple batch files and the Windows scheduling tool. Considerably more complex variations are possible.

---

### Batch File to Start the Standalone Agent

The batch file for starting the Standalone Agent has the following general format:

**Figure 13–30 Example Batch File for Starting Standalone Agent**


```

Start_POG_IMPORT_MASTER.bat - Notepad
File Edit Format View Help
REM #
REM # This batch file is used to run a Planogram Import
Rem #
Rem # This line changes to the required ODI Standalone Agent directory
cd C:\oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin
REM # This line starts the Standalone Agent
start agent -NAME=POG_IMPORT_AGENT

```

1. A line to change the directory to the C:\Oracle\_Base\ODI\_Home\oracledi\agent\bin directory.
2. A line to start the agent. This has the format: **start agent -NAME=POG\_IMPORT\_AGENT**.

This syntax can be deconstructed as follows:

- **start** is a command instructing that the standalone agent be started in a new command window.
- **agent** is the batch file that is used to start standalone agents.
- **-NAME=POG\_IMPORT\_MASTER** is a parameter used to start a specific standalone agent. **-NAME** is the name of the parameter: it is preceded by a '-' sign and should be in capitals. The '=' sign and the information following it is the name of the standalone agent: in this example **POG\_IMPORT\_MASTER**.

### Batch File to Stop the Standalone Agent

The batch file for stopping the Standalone Agent has the following general format:

**Figure 13–31 Example Batch File for Stopping Standalone Agent**


```

Stop_POG_IMPORT_MASTER.bat - Notepad
File Edit Format View Help
REM #
REM # This batch file is used to terminate a Standalone Agent
Rem #
Rem # This line changes to the required ODI Standalone Agent directory
cd C:\oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin
REM # This line stops the Standalone Agent
agentstop -NAME=POG_IMPORT_AGENT

```

1. A line to change the directory to the C:\Oracle\_Base\ODI\_Home\oracledi\agent\bin directory.
2. A line to stop the agent. This has the format: **agentstop -NAME=POG\_IMPORT\_AGENT**.

This syntax can be deconstructed as follows:

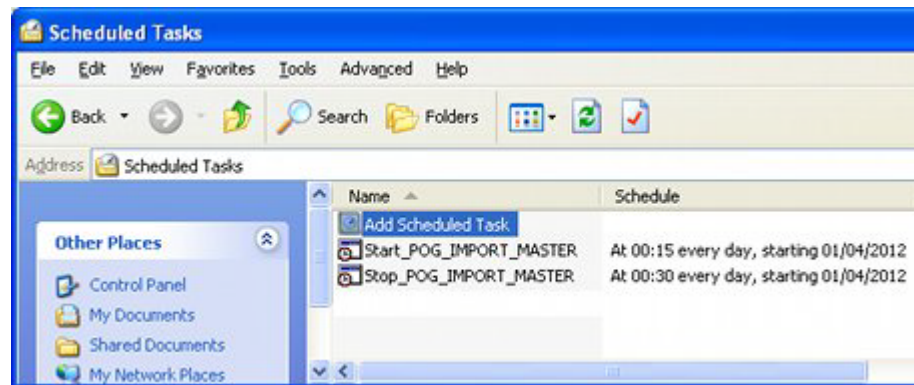
- **agentstop** is the batch file that is used to stop standalone agents.

- `-NAME=POG_IMPORT_MASTER` is a parameter used to stop a specific standalone agent. `-NAME` is the name of the parameter: it is preceded by a `'-'` sign and should be in capitals. The `'='` sign and the information following it is the name of the standalone agent: in this example `POG_IMPORT_MASTER`.

### Scheduling the Standalone Agent

The Standalone agent can be started and stopped to a particular schedule by invoking the batch files from a scheduling tool. A very simplistic way of doing this is through the Windows Scheduling tool.

**Figure 13–32 Scheduling Using Windows Scheduler**



In this simple example, the batch file to start the Standalone Agent is invoked at 15 minutes past midnight and the batch file to stop it at 30 minutes past midnight.





---

---

## Overview of Planogram Import Process

This chapter gives an overview of the logic of the import process. It is not intended to be a full technical description, just an aid to understanding any reported errors.

### User Determined Parameters Affecting Import Process

This section gives a brief explanation of the main user determined parameters affecting the planogram import process.

---

---

**Note:** For more information on how to set these parameters see the chapter on running the import process.

---

---

#### When Imports are Executed

Planogram Imports are executed in one of three cases:

- **A manual import initiated from ODI Studio**

This is a manually initiated import from the Operator Navigator in ODI Studio. After the import has been started, all XML files currently in the specified import directory will be processed. The import will then terminate.

- **A scheduled import executed by a local or standalone agent**

This is an import initiated by a local or standalone agent based on a schedule specified within ODI. Import will start and stop at the specified times, with all XML files in the specified import directory being processed.

Scheduled imports require the agent to be running permanently.

- **An import initiated by a scheduling tool**

This is an import that will start and stop as the agent is started and stopped by an external scheduling tool. After the import has been started, all XML files currently in the specified import directory will be processed. The import will then terminate at the specified time.

Imports using a scheduling tool generally only have the agent running at specified times.

#### Parameters Affecting an Import

Three parameters can affect an import. These are specified when an import is manually initiated, scheduled or run via a scheduling tool. These are:

- **Planogram Import Date**

This option is set so that only planograms with a last modified date later than a specified date are imported.

- **Overwrite Option**

This option is set so that existing data in the database is overwritten if data exists in the information being imported.

- **Polling Interval**

The polling option allows the import directory to be rechecked at specified interval during the time the import is active. This polling interval can be set at intervals ranging from 0 - 60 minutes. If set to 0, the import directory will only check when the import process is started.

**Level of Detail Imported**

The level of detail imported is governed by the nodes included in the XML file containing the planogram information. This is generated by the Source Bridge and can contain information ranging from summary information on the planogram (header information to full details on equipment, products and planogram performance.

**Types of Logging**

Data relating to the import is logged in the following places:

- **ODI**

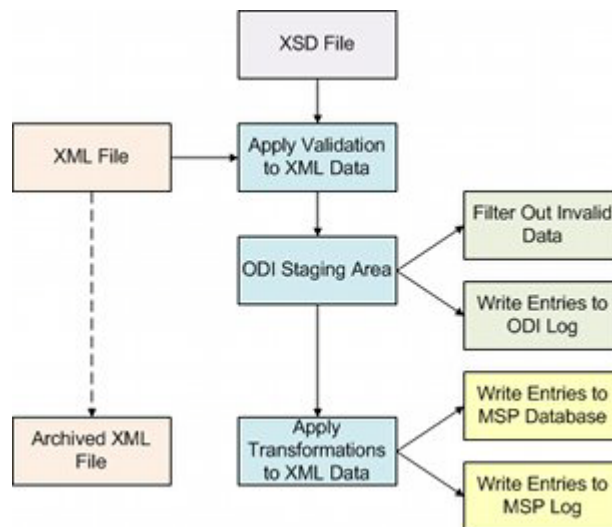
Data relating to the transfer of information will be written to the ODI work Repository and can be read using the Operator Navigator in ODI Studio.

- **MSP Database**

Log entries relating to the values of the data (for example where defaults or nulls have been applied) are written to the AVTTB\_ODI\_LOG table.

## Operating Logic

*Figure 14–1 Operating Logic*



### Read Values from MSP Database

The first stage of the import process is to read values from the Macro Space Planning (MSP) database. These include:

- Location of Import and Archive folders
- System Variables giving defaults for imported values.

These system variables will be referenced during the planogram import.

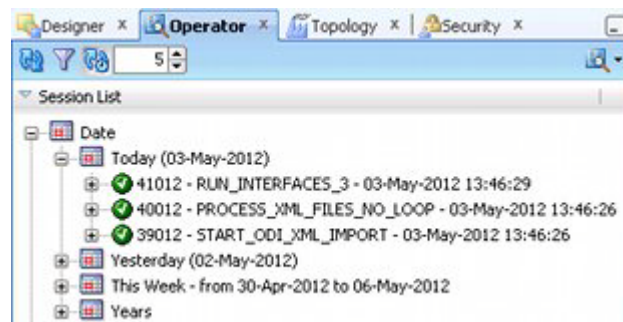
### Prepare List of XML Files for Import

The Import folder specified by the **DEFAULT\_IMPORT\_DIR** system variable is scanned and a list of files generated. The files will be ordered in alphabetical order and processed in that sequence.

### ODI Logging

Logging will commence as soon as processing begins. The results relating to the transfer of data are written to the ODI logs. These are in the Operator Navigator in the Session List Accordion and will steadily be added to as the import proceeds.

**Figure 14–2 Operator Navigator - Log Results in Session List Accordion**



### Load to Staging Area

For each XML file to be processed, the following actions will be carried out:

1. The XSD file (located in a specified directory that is specified within ODI Studio) is read and a set of blank tables will be created in the ODI Staging Area.
2. The Root Element in the XML files is checked. If this is not **MSM\_IMPORT\_DATA\_STRUCTURE** the file will be considered invalid and no data will be loaded into the ODI Staging Area.
3. The data from the XML file is loaded to the ODI Staging Area. Some XML files may only contain a subset of data, for example Product and Equipment definitions.

### Validating the Data

Once loaded into the ODI Staging Area, the data will be validated if the **POG\_IMPORTER\_FILTER\_OPTION** system variable in the MSM Administration module is set to 1 (on) This is done as follows:

- All Mandatory tags must be present (see Appendix C for details). If a mandatory tag is missing, the record will be rejected.
- If any optional tags are missing, they will be loaded as **Null** or **Default** values.

### Transforming and Importing the Data

Once all data has been validated, it will be brought into the ODI Staging area. It is then transformed and imported into the MSP database. Data is imported in the following sequence:

1. Equipment Definitions
2. Product Definitions
3. Planogram Hierarchy
4. Planogram

All Scenarios in ODI are executed. If one of those scenarios references a table that is blank in the ODI staging area, no data is imported into the MSP Database.

Transformations will be applied during the import. Data is converted from the form it is held in the third party planogram design software to that one that conforms to that required by the MSP database.

### MSP Logging

Log entries related to the data itself are written to the AVTTB\_ODI\_LOG table.

**Figure 14–3 AVTTB\_ODI\_LOG Table - Example Results**

ODL_ID	ODL_TEXT	ODL_START_DATE
701	GENERAL - Starting ODI Import	03-MAY-2012 12:00:00
718	WARNING - Using Null value for BRAND	03-MAY-2012 01:46:44
719	WARNING - Using default value for colour.	03-MAY-2012 01:46:44
720	WARNING - Using Null value for MANUFACTURER	03-MAY-2012 01:46:44
721	WARNING - Using Active as default Status Code.	03-MAY-2012 01:46:44
722	WARNING - Using Null value for SUPPLIER	03-MAY-2012 01:46:44
723	WARNING - Using Null value for IMPORTNAME	03-MAY-2012 01:46:44
724	WARNING - Using Default Value for IMPORTDATE	03-MAY-2012 01:46:44

As can be seen from the above example, the table has logged when the import started, followed by a list of values where nulls or defaults have been assigned.

### Archiving the Files

After the XML File has been processed, it is moved from the Import Directory to the Archive Directory. A timestamp is added to the file name to indicate when the file was processed.

The intent behind moving the files after processing is to prevent the situation where a file containing errors remains in the Import Directory and is repeatedly processed if the ODI settings are such that the Import directory is polled for fresh files at regular intervals.

---

---

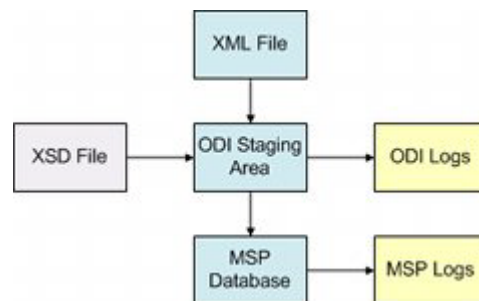
## Using Planogram Import Logs

This chapter explains how to use the planogram import logs to review the import results and to identify any resulting problems.

### Overview of using Import Logs

There are two forms of Import Logs; those associated with data being processed by Oracle Data Integrator (ODI) and those associated with data being inserted into the Macro Space Planning (MSP) database.

*Figure 15-1 Planogram Import Process Logging*



#### **ODI**

Data is imported into the ODI Staging Area using tables created referencing the XSD file. Data pertinent to populating these tables and transforming the data is written to the ODI logs.

#### **MSP Database**

After data has been moved to the ODI Staging area, it is next imported into the MSP database. Information on planogram data modified during this process is written to the AVTTB\_ODI\_LOG table.

### ODI Log Information

This section contains overview information on ODI Log information.

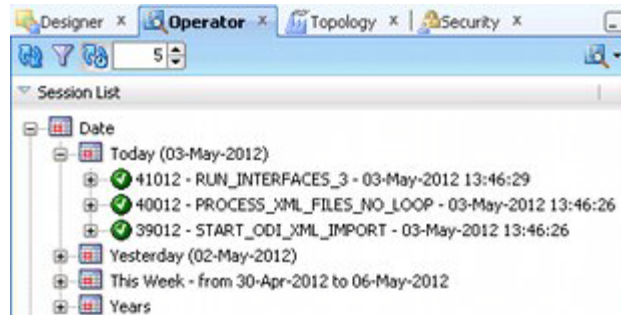
#### Accessing the ODI Log information

ODI Log information is accessed as follows in ODI Studio:

1. Connect to the Work Repository. This is done from the ODI menu.

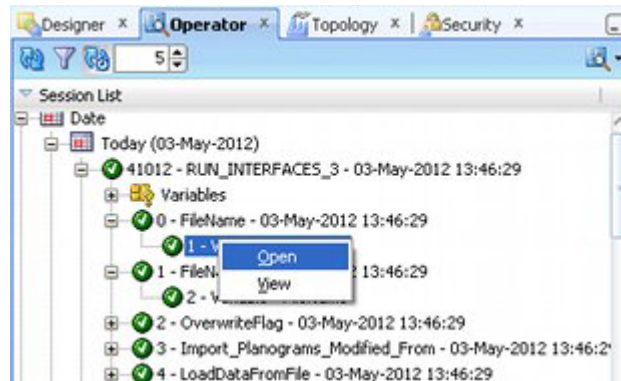
2. Open the Operator Navigator.
3. Open the Session List Accordion by clicking the small triangle to the left of the name.

**Figure 15–2 Operator Navigator - Session List Accordion**



4. Log results from imports will be stored in date sequence. Open the pertinent part of the hierarchy.
5. Specific log entries are accessed by highlighting them in the hierarchy, right clicking and selecting **Open**.

**Figure 15–3 Operator Navigator - Opening Log Entry**



6. The specific entry in the ODI log will then display.

Figure 15–4 Example ODI Log Entry

- The **Definition Tab** gives details of the task, together with an error message.
- The **Code Tab** gives details of the code segment that the action is associated with. The code referenced is supplied in compiled form so cannot be directly accessed on site. However, this information may be helpful if support is required from Oracle.
- The **Connection Tab** give details of where the source data is being read from, together with where the data is to be imported to (target database).
- The **Privileges Tab** is related to ODI. For more information, see the ODI documentation.

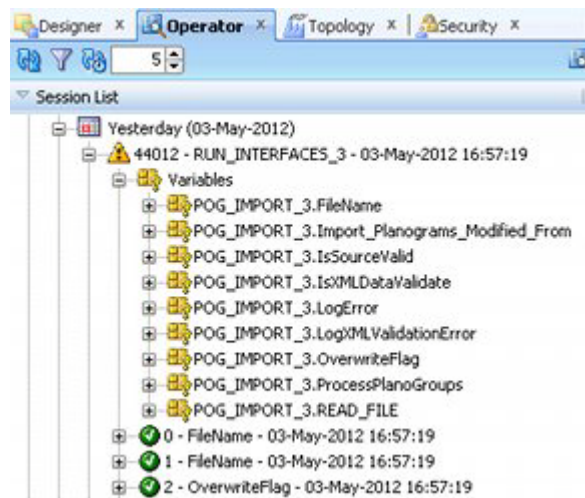
## Log Information Results

This section gives additional information on the ODI Log results.

### Variables and Log Results

Within the ODI Logs, there are two types of data: Variables and Results.

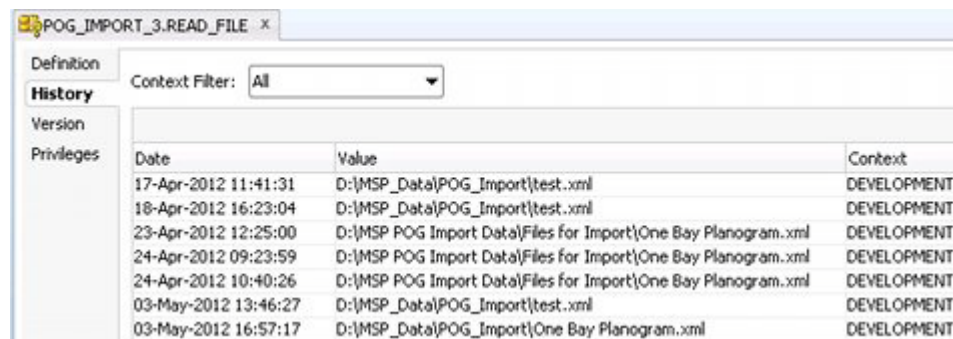
**Figure 15–5 ODI Log - Variables and Results**



**Variables**

Variables hold the results of parameters used during the import process. They are indicated by a square icon. These values are displayed by highlighting the variable, right clicking and selecting Open.

**Figure 15–6 ODI Log - Example of Variable Values**

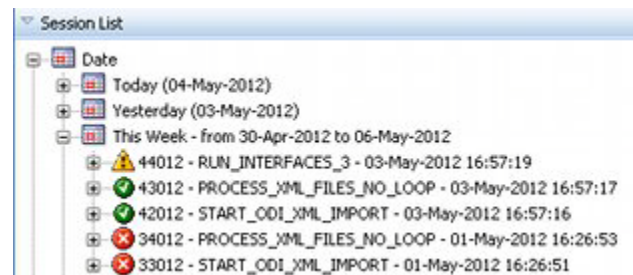


In the above example, the POG\_IMPORT\_3.READ\_FILE variable holds information on all XML files that have been processed recently.

**Results**

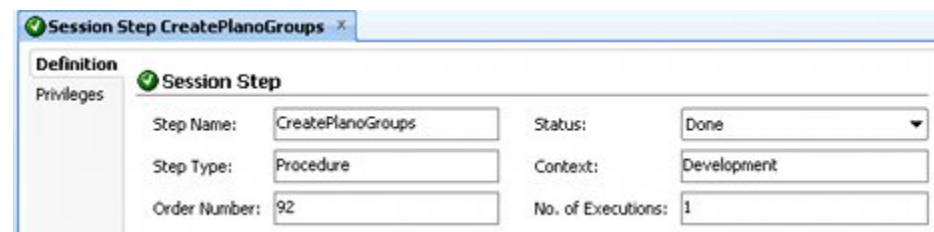
Results hold the outcome of each processing stage during the import. They are indicated by circular or triangular symbols.



**Figure 15–7 Types of ODI Log Messages**

- **Green Circle with Tick:** This indicates the action has completed.
- **Yellow Triangle with Explanation Mark:** This indicates an action with warnings.
- **Red Circle with Cross:** This indicates an action with an ODI error flag.

The details associated with each result can be seen by highlighting the result, right clicking and selecting Open.

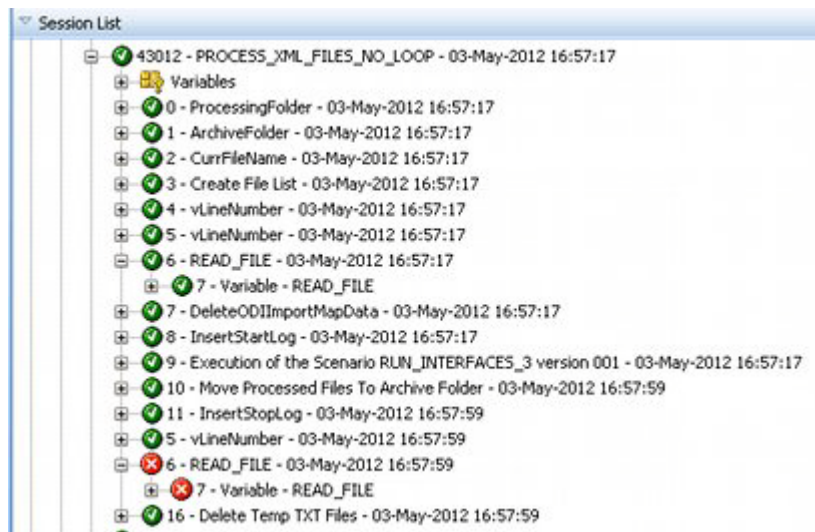
**Figure 15–8 ODI Log - Example of Results**

In the above example the CreatePlanoGroup (Create Planogram Group) step can be seen to have executed correctly.

### Error Result on End of Run

Some error messages do not indicate actual errors in the processing of the data, but that ODI cannot execute as task.

**Figure 15–9 Error Associated With No Further Files To Process**



In the above example, the first log entry for 6 - READ\_FILE is green, indicating successful processing. The second log entry is red, indicating ODI has failed to execute a task. Inspection of the log entry will reveal the reason.

**Figure 15–10 Error Message: 6 - READ\_FILE**



The error message shows that the import process has failed to find a file name to process. However, in this case it is because ODI has processed all available XML files. It is thus necessary to evaluate error message to establish the situation that resulted in them being generated.

## Macro Space Planning Log Information

After data has been imported into MSP, the results are written to two tables:

- **AVTTB\_ODI\_LOG table:** this holds the results of all data that was modified or updated during a planogram import.
- **SNP\_CHECK\_TAB table:** this holds the results of all data that was rejected during the planogram import process. This table is created by ODI when the first import is run.

## AVTTB\_ODI\_LOG Table

Information in the AVTTB\_ODI\_LOG table is stored as follows:

**Figure 15–11 Columns in AVTTB\_ODI\_LOG Table**

COLUMN_NAME	DATA_TYPE	NULLABLE
ODL_ID	NUMBER	No
ODL_TEXT	VARCHAR2(4000 BYTE)	No
ODL_START_DATE	DATE	No
ODL_TYPE	NUMBER	Yes
ODL_STOP_DATE	DATE	Yes
ODL_FILENAME	VARCHAR2(4000 BYTE)	Yes

- **ODL\_ID**  
This field is the primary key for the table. It will increment by 1 for each entry.
- **ODL\_TEXT**  
This field holds the text of the error message.
- **ODL\_START\_DATE**  
This field holds the start date and time for this part of the planogram import. this field will populate for all operations, showing the time the operation was commenced.
- **ODL\_TYPE**  
This field holds a flag showing the type of error.
  - 1 = Error
  - 2 = Success
  - 3 = Warning
  - 4 = General
- **ODL\_STOP\_DATE**  
This field holds the stop date and time for this part of the planogram import. It will only populate if the specific operation referenced by the row is stopped.
- **ODL\_FILENAME**  
This field holds the path to and filename of the XML file that is being used for this specific planogram import. It will only populate at the start and end of the import.

### Accessing Data from the AVTTB\_ODI\_LOG Table

Accessing data from the AVTTB\_ODI\_LOG table is achieved in different ways:

- Directly accessed from the database using a tool like SQL Developer.
- A customer specific report developed for a reporting tool.

### Purging the AVTTB\_ODI\_LOG Table

It is recommended that a policy be developed for purging unwanted data from the AVTTB\_ODI\_LOG table. This will prevent the table growing to an undue size and impacting on planogram import performance.

## SNP\_CHECK\_TAB Table

Information in the SNP\_CHECK\_TAB table is stored as follows:

**Figure 15–12 Columns in SNP\_CHECK\_TAB Table**

COLUMN_NAME	DATA_TYPE	NULLABLE
CATALOG_NAME	VARCHAR2(100 CHAR)	Yes
SCHEMA_NAME	VARCHAR2(100 CHAR)	Yes
RESOURCE_NAME	VARCHAR2(100 CHAR)	Yes
FULL_RES_NAME	VARCHAR2(100 CHAR)	Yes
ERR_TYPE	VARCHAR2(1 CHAR)	Yes
ERR_MESS	VARCHAR2(250 CHAR)	Yes
CHECK_DATE	DATE	Yes
ORIGIN	VARCHAR2(100 CHAR)	Yes
CONS_NAME	VARCHAR2(35 CHAR)	Yes
CONS_TYPE	VARCHAR2(2 CHAR)	Yes
ERR_COUNT	NUMBER(10,0)	Yes

- **CATALOG\_NAME**  
 This field is not used for planogram import errors.
- **SCHEMA\_NAME**  
 This is the name of the MSP schema within the database instance that data is being imported into.
- **RESOURCE\_NAME**  
 This is the name of the table within the MSP schema.
- **FULL\_RES\_NAME**  
 This is the formalized name of the table in the form **<schema name.table name>**.
- **ERR\_TYPE**  
 This field holds a code identifying the type of the error.
- **ERR\_MESS**  
 This field holds the error message associated with the error.
- **CHECK\_DATE**  
 This field holds the date and time the error was generated.
- **ORIGIN**  
 This field identifies the ODI scenario and action within the scenario associated with the error.
- **CONS\_NAME**  
 This field identifies the constraint in the table associated with the error. It may be a field name, primary key, foreign key, etc.
- **CONS\_TYPE**  
 This field holds a code identifying the type of error associated with the constraint in the CONS\_NAME field.
  - FK: Foreign Key
  - NN: Column

- PK: Primary Key
- **ERR\_COUNT**  
This field holds the number of errors associated with a specific error. There will be a minimum of one.

#### **Accessing Data from the SNP\_CHECK\_TAB Table**

Accessing data from the SNP\_CHECK\_TAB table is achieved in different ways:

- Directly accessed from the database using a tool like SQL Developer.
- A customer specific report developed for a reporting tool.

#### **Purging the SNP\_CHECK\_TAB Table**

This table will be cleaned up by ODI on each iteration of an ODI Planogram Import.

### **Collation of Log Information**

Log information can be grouped to allow all log messages for a single planogram import to be retrieved together. This is done by the parent node, element and the object being imported being included in the log table for each message. This allows log messages to be grouped for each import object; for example all log messages for the 12 foot confectionery planogram.



---

---

## Troubleshooting Initial Problems

This section of the documentation gives solutions to commonly experienced problems when setting up and testing the ODI Planogram Import solution.

### General Issues

Some issues are related to limitations of the functionality.

#### Planogram Import does not currently handle placeholders

Planograms can be specified in a variety of different formats.

*Figure 16–1 Planogram Variants*



In the above screen shot there are three forms:

- **2D Form**

This is shown on the left. It is essentially 'header' information where the planogram is shown of being a specific type and occupying a specific volume.
- **Placeholder Form**

This is shown in the centre. Here the shelves are included in the planogram data, but the products are shown as placeholders indicating the general type of merchandiser without information on SKU or quantity.
- **3D Form**

This is shown on the right. Here the planogram is specified in full detail with information on both equipment and merchandise completely specified.

The ODI Planogram Import solution can import planograms in both 2D and 3D form. It is not presently capable of importing planograms in placeholder form.

## Installing and Configuring ODI

These are some of the common problems experienced when installing and configuring ODI.

### Dynamic and Static IP Addresses

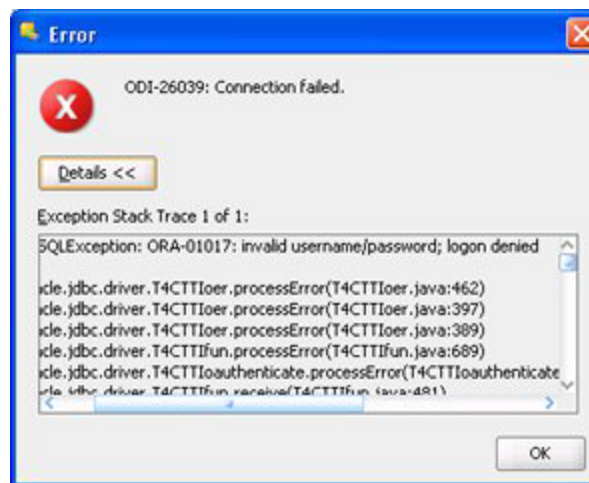
Computers can have either static or dynamic IP addresses. If a computer has a static IP address, that address will not change until manually changed by a system administrator. If a computer has a dynamic IP address, that IP address can be automatically changed by the Dynamic Host Control Protocol (DHCP) server within a large retail organization.

Although Oracle databases can be configured to cater for dynamic IP addresses, it is simpler to ensure any machines hosting those databases have static IP addresses.

### Unable to Connect to Repository

This problem occurs during installation when incorrect passwords are supplied. It is often found when the **Test Connection** button is clicked when setting up a Master or Work Repository.

**Figure 16–2** Connection Failed warning



The solution for this problem is to verify that the correct passwords are being used. [Appendix: Installation Detail Example](#) has been supplied as a method of minimizing this problem.

### Repository Already Exists Warning when Creating Master Repository

When importing the Master Repository, if a Repository ID of 1 is entered into the Master Repository Import Wizard, an error may result saying Master Repository already exists.



**Figure 16–3 Master Repository Exists Warning**

This is because the Master Repository being imported has a Repository ID of 1. The solution is to enter a Repository ID of greater than 1.

---

**Caution:** This circumstance only exists when configuring ODI Studio for the first time. If ODI Studio is being reinstalled, ensure that you are not overwriting an existing Master Repository.

---

## Old Login Information Exists After Reinstalling ODI

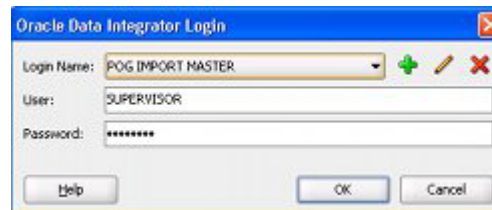
If ODI is removed and reinstalled on a computer, a list of repositories from the old installation is still retained. This means errors such as 'Repository name already exists' may occur when configuring the new repositories. Additionally, when selecting a repository to connect to, the repositories from the previous installation may be included in the list of available repositories.

### Where information is stored

The repository connection information is stored in an XML file called **snps\_login\_work.xml**. This can be found at **C:\Documents and Settings\<your username>\Application Data\odi\oracledi**. If necessary, information on old repositories can be removed from this file.

### Removing old repository information

The simplest way of storing the old repository entries is to delete them using the ODI Login dialog box.

**Figure 16–4 ODI Login Dialog Box**

To delete the old entries:

1. Select the old login from the Login Name drop down list.
2. Click the Delete icon.

## ODI Studio Functionality

These are practical problems that could be experienced with the ODI functionality.

### Navigator disappears from View

If any of the ODI Studio Navigators is inadvertently dragged beyond the bounds of the window, it disappears from view and cannot be restored via the ODI studio menus.

The positions of the navigators (and other ODI information) is stored in a file called **windowinglayout.xml**. This can be found at in the application data folder. A typical path is: **C:\Documents and Settings\\Application Data\odi\system11.1.1.6.0\o.ide.11.1.1.6.38.61.92**. Correcting the data in the XML file will restore the navigator.

---

**Note:** Please consult an XML expert for information as to how to correct the parameters in this file.

---

### Running an Import

These are common problems experienced when running a planogram import.

### Location of Import and Archive Folders

The functionality starting the data transfer requires the Import and archive folders holding the XML files with planogram information to be local to it. For example, if the planogram import is manually initiated from ODI Studio, the functionality will expect to find the pertinent folders on the computer hosting ODI Studio. Conversely, if the import is initiated by a standalone agent on a server, the pertinent folders will be expected to be on that folder.

---

**Note:** The location where the import is initiated from may well vary depending on whether the import is being run for testing or production purposes.

---

If the files are not in the expected location, an error message will result when the import is run.

**Figure 16–5 Typical Operator Navigator Error Message for Wrong File Location**

```

Variable and Sequence Values
Error Message:
ODI-1226: Step READ_FILE fails after 1 attempt(s).
ODI-1228: Task READ_FILE (Variable) fails on the target connection FileNameListDataServer.
Caused By: java.sql.SQLException: ODI-40438: File not found: D:\MSP_Data\POG_Import\All_Files_Names.txt
at com.sunopsis.jdbc.driver.file.FileResultSet.<init>(FileResultSet.java:162)
at com.sunopsis.jdbc.driver.file.impl.commands.CommandSelect.execute(CommandSelect.java:57)
at com.sunopsis.jdbc.driver.file.CommandExecutor.executeCommand(CommandExecutor.java:33)
at com.sunopsis.jdbc.driver.file.FilePreparedStatement.executeQuery(FilePreparedStatement.java:135)
at oracle.odi.query.JDBCTemplate.executeQuery(JDBCTemplate.java:189)

```

The solution for this problem is to create the pertinent folders on each machine that might be used to initiate a planogram import. The system variables holding the locations of these files (**DEFAULT\_IMPORT\_DIR** and **DEFAULT\_IMPORT\_**

ARCHIVE\_DIR) must be changed in the MSM Admin Module prior to changing the machine the planogram import is initiated on.

## ODI Staging Area Schema Name Not Exactly Seven Characters

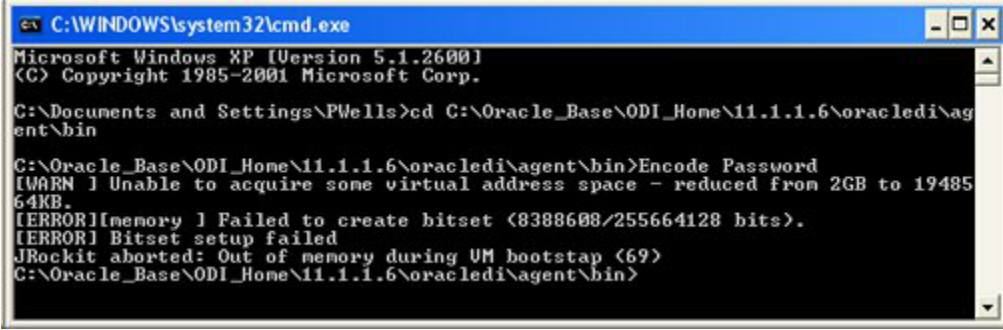
The ODI Staging Area schema name must be exactly seven characters long. This is because Oracle databases truncate table names that exceed 30 characters. Because Oracle databases truncate table names in a predictable manner, having the ODI Staging Area name exactly seven characters long enables the functionality to cater for truncated table names.

Failure to keep to a seven character name will lead to errors during data import.

## Lack of Memory Error (Java Heap Size Too Big)

The minimum and maximum Java Heap sizes are stored in the `odiparams.bat` batch file found in the `C:\Oracle_Base\ODI_Home\oracledi\agent\bin` directory found where an agent has been installed. This holds a parameter called `ODI_MAX_HEAP`. If this parameter is set too high relative to the memory on the computer the `odiparams.bat` file is installed on, an out of memory error will result.

**Figure 16–6 Error from Java Heap Size Being Too Large**



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\PWells>cd C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin

C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>Encode Password
[WARN ] Unable to acquire some virtual address space - reduced from 2GB to 1948564KB.
[ERROR][memory ] Failed to create bitset (8388608/255664128 bits).
[ERROR] Bitset setup failed
JRockit aborted: Out of memory during VM bootstrap (69)
C:\Oracle_Base\ODI_Home\11.1.1.6\oracledi\agent\bin>

```

In the above example the `encode.bat` file has been run to demonstrate the problem. A typical error message has resulted.

The solution is to set the `ODI_MAX_HEAP` parameter to one compatible for the machine that will run the agent.

## Path to XSD file must be specified in full for XML configuration

The XSD file is used during planogram import to create temporary tables in the ODI Staging Area. If it cannot be found, the temporary tables cannot be found and the planogram import will fail. This will result in an error message such as the one below.

**Figure 16–7 Error Message - Incorrect Path to XSD File**

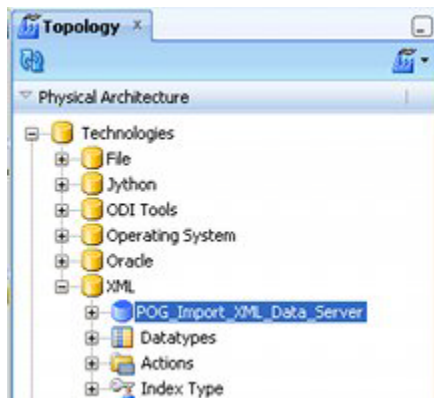
```

Error Message:
ODI-1228: Task READ_FILE (Variable) fails on the target connection FileNameListDataServer.
Caused By: java.sql.SQLException: ODI-40438: File not found: D:\MSP POG Import Data\Files for Import\All_Files_Names.txt
at com.sunopsis.jdbc.driver.file.FileResultSet.<init>(FileResultSet.java:162)
at com.sunopsis.jdbc.driver.file.impl.commands.CommandSelect.execute(CommandSelect.java:57)
at com.sunopsis.jdbc.driver.file.CommandExecutor.executeCommand(CommandExecutor.java:33)
at com.sunopsis.jdbc.driver.file.FilePreparedStatement.executeQuery(FilePreparedStatement.java:135)
at oracle.odi.query.JDBCTemplate.executeQuery(JDBCTemplate.java:189)
    
```

This problem can occur if the path specified to the XSD file is a local path and the planogram import has been initiated from another computer.

The solution is to ensure the path to the XSD file is specified in full. This is done in the Physical Architecture Accordion of the Topology Navigator. (See the chapter on configuring ODI Studio for full information).

**Figure 16–8 Topology Tab - Physical Architecture - XML**



## Appendix: Installation Detail Example

The installation of the ODI Planogram Import solution involves the creation of a number of schemas and directories, together with the placement of a number of files. These operations may be carried out by a number of different individuals. This appendix can be used to note the details to assist the installation process.

### Appendix: Installation Detail Example

The following tables hold the information used in the worked example used for this documentation.

**Table A-1 Database Instance Connection Details**

Database Instance Connection Details	
This information is for the database instance holding the MSP Schema, Master Repository, work Repository and ODI Staging Area.	
<b>Note:</b> these schemas are commonly in the same database instance. This reduces latency.	
Server Name	Specify name of your server
Database Instance Name	SPACEPLAN
User (DBA Privileges)	system
Password	Password_Spaceplan
Listener Port	1521
Oracle SID	SPACEPLAN

**Table A-2 ODI Master Repository Configuration Details**

ODI Master Repository Connection Details	
Server Name	Specify name of your server
Database Instance Name	SPACEPLAN
Master Repository Name (Schema Name)	POG_IMPORT_MASTER
Master Repository User	POG_IMPORT_MASTER
Master Repository User Password	Password_Master
User (DBA Privileges)	system
DBA Password	SPACEPLAN
Listener Port	1521

**Table A-2 (Cont.) ODI Master Repository Configuration Details**

Oracle SID	SPACEPLAN
Repository Number	2 or greater

**Table A-3 ODI Work Repository Configuration Details**

ODI Work Respository Connection Details	
Server Name	Specify name of your server
Database Instance Name	SPACEPLAN
Master Repository Name (Schema Name)	Pog Import Master
Master Repository User	POG_IMPORT_MASTER
Master Repository User Password	Password_Master
User (DBA Privileges)	system
DBA Password	Password_Spaceplan
ODI Password (10 characters max)	Work_Rep [Not otherwise used during implementation].
Listener Port	1521
Oracle SID	SPACEPLAN
Repository Number	Must be different from Master Repository

**Table A-4 ODI Staging Area Connection Details**

ODI Staging Area Connection Details	
Server Name	Specify name of your server
Database Instance Name	SPACEPLAN
User (DBA Privileges)	system
DBA Password	Password_Spaceplan
Listener Port	1521
Oracle SID	SPACEPLAN
Staging Area User Name (Must be seven characters)	XML HOLD
Staging area Password	Password_XMLHOLD

**Table A-5 Planogram Import Directory Details**

Planogram Import Directory Details	
<b>Note:</b> This directory must be local to the ODI object (Local or Standalone Agent) invoking the planogram import.	
Computer Name	[Specify name of computer of server currently hosting planogram import directory]
Directory Name and Path	\\<machine name>\MSP_Data\POG_Import

**Table A-6 Planogram Archive Directory Details**

Planogram Archive Directory Details	
<b>Note:</b> This directory must be local to the ODI object (Local or Standalone Agent) invoking the planogram import.	
Computer Name	Specify name of computer of server currently hosting planogram archive directory
Directory Name and Path	\\<machine name>\MSP_Data\POG_Archive

**Table A-7 Planogram XSD Directory Details**

Planogram XSD Directory Details	
<b>Note:</b> This Directory holds the XSD file used to create the tables in the ODI staging area prior to each planogram import. It is referenced by ODI.	
Computer Name	Specify name of computer of server currently hosting planogram XSD directory
Directory Name and Path	\\<machine name>\MSP_Data\POG_XSD

**Table A-8 ODI Login Details**

ODI Login Details	
<b>Note:</b> These are the default user name and password for ODI.	
User Name	SUPERVISOR
Password	SUNOPSIS

**Table A-9 ODI Standalone Agent Connection Details**

ODI Standalone Agent Details	
Server Name	Specify name of computer of server currently hosting standalone agent
Stand Alone Agent Name	POG_IMPORT_AGENT
Standalone Agent Port	20910
Oracle Home Location	Depends where installed

**Points to Watch**

- When running an import, the directories used for planogram import and archiving must be local to the machine initiating the import.
- Ensure that a copy of the **ExtXML.properties** file exists in **C:\Oracle\_Base\Oracle\_Home\oracledi\client\jdev\extensions\oracle.odi.navigator\lib** directory.





## Appendix: Installation Detail Checklist

The installation of the ODI Planogram Import solution involves the creation of a number of schemas and directories, together with the placement of a number of files. These operations may be carried out by a number of different individuals.

This appendix can be used to note the details of passwords, file locations and other pertinent information to assist the installation process.

### Appendix: Installation Detail Checklist

**Table B-1 Database Instance Connection Details**

Database Instance Connection Details	
This information is for the database instance holding the MSP Schema, Master Repository, work Repository and ODI Staging Area.	
<b>Note:</b> these schemas are commonly in the same database instance. This reduces latency.	
Server Name	
Database Instance Name	
User (DBA Privileges)	
Password	
Listener Port	
Oracle SID	

**Table B-2 ODI Master Repository Configuration Details**

ODI Master Repository Connection Details	
Server Name	
Database Instance Name	
Master Repository Name (Schema Name)	
Master Repository User	
Master Repository User Password	
User (DBA Privileges)	
DBA Password	
Listener Port	
Oracle SID	

**Table B-2 (Cont.) ODI Master Repository Configuration Details**

Repository Number	
-------------------	--

**Table B-3 ODI Work Repository Configuration Details**

ODI Work Repository Connection Details	
Server Name	
Database Instance Name	
Master Repository Name (Schema Name)	
Master Repository User	
Master Repository User Password	
User (DBA Privileges)	
DBA Password	
ODI Password (10 characters max)	
Listener Port	
Oracle SID	
Repository Number	

**Table B-4 ODI Staging Area Connection Details**

ODI Staging Area Connection Details	
Server Name	
Database Instance Name	
User (DBA Privileges)	
DBA Password	
Listener Port	
Oracle SID	
Staging Area User Name (Must be seven characters)	
Staging area Password	

**Table B-5 Planogram Import Directory Details**

Planogram Import Directory Details	
<b>Note:</b> This directory must be local to the ODI object (Local or Standalone Agent) invoking the planogram import.	
Computer Name	
Directory Name and Path	

**Table B-6 Planogram Archive Directory Details**

Planogram Archive Directory Details	
<b>Note:</b> This directory must be local to the ODI object (Local or Standalone Agent) invoking the planogram import.	

**Table B-6 Planogram Archive Directory Details**

Computer Name	
Directory Name and Path	

**Table B-7 Planogram XSD Directory Details**

<b>Planogram XSD Directory Details</b>	
<b>Note:</b> This Directory holds the XSD file used to create the tables in the ODI staging area prior to each planogram import. It is referenced by ODI.	
Computer Name	
Directory Name and Path	

**Table B-8 ODI Login Details**

<b>ODI Login Details</b>	
<b>Note:</b> These are the default user name and password for ODI.	
User Name	
Password	

**Table B-9 ODI Standalone Agent Connection Details**

<b>ODI Standalone Agent Details</b>	
Server Name	
Stand Alone Agent Name	
Standalone Agent Port	
Oracle Home Location	

**Points to Watch**

- When running an import, the directories used for planogram import and archiving must be local to the machine initiating the import.
- Ensure that a copy of the **ExtXML.properties** file exists in **C:\Oracle\_Base\Oracle\_Home\oracledi\client\jdev\extensions\oracle.odi.navigator\lib** directory.



---

---

## Appendix: Mandatory XML Elements

This appendix specifies the mandatory elements that must be present in the XML file. If any of these mandatory elements are missing, the record will be rejected and an error logged. The import process will then attempt to process the next record.

### Appendix - Mandatory XML Elements

#### Option Node

The Options node and all of the sub-nodes and elements are compulsory in the planogram input file. If the nodes are not found then an error will be logged and the planogram import ODI scenario will end without importing any planograms.

#### Product Node

The following elements are mandatory in the Product Node:

- Name
- Description
- Code

#### Display Style Node

This note is a child of the product node. The node itself is mandatory for the Product Node. It will only be included once per Product Node. The following elements are mandatory:

- Type
- Length
- Depth
- Height

#### Equipment Node

The following elements are mandatory in the Equipment Node:

- ModelFilename
- Equipment Type
- Code
- EquipmentStyle
- Length

- Depth
- Height

### **Planogram Group**

The following elements are mandatory in the Planogram Group Node:

- Name
- Group Code

### **Planogram Node**

The following elements are mandatory in the Planogram Node:

- Name
- ClientCode
- Length
- Depth
- Height

### **Planogram Bay Node**

The following elements are mandatory in the Planogram Bay Node:

- Length
- Depth
- Height
- VerticalPosition
- DepthPosition
- HorizontalPosition
- Roll
- Pitch
- Yaw

### **Equipment Positions Mode**

The following elements are mandatory in the Equipment Position node:

- Length
- Depth
- Height
- VerticalPosition
- DepthPosition
- HorizontalPosition
- Roll
- Pitch
- Yaw

**Product Position Node**

The following elements are mandatory in the Product Position Node:

- Code
- DisplayStyleType
- Length
- Depth
- Height

**Labels Node**

The following elements are mandatory in the Labels Node:

- Length
- Depth
- Height
- HorizontalPosition
- DepthPosition
- VerticalPosition
- Pitch
- Yaw





# Glossary

This glossary describes the terms used in data import via Oracle Data Integrator (ODI).

## **2D Planogram**

Planogram design containing summary information only: this typically includes the name and code of the planogram, together with information on the fixture size and number of bays.

## **3D Planogram**

Planogram design containing full information including the position of shelves and products, together with financial information and User Defined Attributes (UDAs).

## **Accordion**

Accordion is a term used for the dividers used within an ODI Navigator.



In the above example, the Topology Navigator contains Physical Architecture, Contexts, Logical Architecture, Languages, Repositories and Generic Action Accordions.

**Agent**

Lightweight Java Process that are used to run imports without the need to have ODI Studio (or ODI Console) active.

**Bay**

Bay is a term used to describe a fixture in a multi-fixture planogram. A planogram designed to occupy four fixtures is often called a four bay planogram.

**Context**

When setting up a specific data import, ODI will take data from a server or file with a specified structure and move it to a server or file with a specified structure. This can happen in different situations; for example Test or Production. Each Context is used to specify the combination of source and target for each format of data import.

**Fitting**

An item of equipment that does not hold merchandise. Fittings can be placed on the floor (for example legs) or go on other items of equipment (for example shelf dividers).

**Fixel**

Another term for shelf object.

**Fixture**

A floor standing item of equipment that can hold merchandise.

**In-Store Space Collaboration**

Web enabled application within the Macro Space Planning Suite. Contains a limited suite of functionality enabling users to edit existing store plans, but not to create new ones.

**ISSC**

Abbreviation for In-Store Space Collaboration.

**Local Agent**

An Agent installed on the same machine as ODI Studio (or ODI Console).

**Macro Space Management**

Application within the Macro Space Planning Suite. Contains the full functionality required for store planning/space measurement requirements. Designed to run on a LAN.

**Macro Space Planning**

Umbrella term for the application suite containing Macro Space Management and In-Store Space Collaboration.

**Master Repository**

Part of a database instance used by ODI to store high level information on the source information and target database.

**MSM**

Abbreviation for Macro Space Management.

**MSP**

Abbreviation for Macro Space Planning.

## Navigator

A Navigator is a module within ODI. There are four navigators - Security, Topology, Designer and Operator. Each controls a different aspect of ODI's operations.



In the above example, the Topology Navigator is open, while the Security, Designer and Operator Navigators are minimized.

## ODI

Abbreviation for Oracle Data Integrator.

## Oracle Data Integrator

An Oracle application used for extracting data from one source and importing into another database.

## Planogram

A report specifying the arrangement of products, shelves and signage on one or more fixtures in a store. Designed to present merchandise in an attractive way to customers.

## Planogram Revision

A planogram revision occurs when the changes made to an existing planogram are minor and the planogram is not saved as a new design. The planogram revision number is generally incremented by one - for example the planogram becomes Revision 2.

## Planogram Revision Change

Process that happens within MSP when planogram revisions in a floor plan are updated to conform to the latest revision.

## Planogram Substitution

Process that happens within MSP when one planogram design is automatically changed for another. An example would be 2 Bay 'Tinned Pea' planogram being changed for a 2 Bay 'Baked Bean' planogram.

## Profile (Planogram)

Profile is an alternative name for a bay in a planogram. This term tends to be MSP specific.

## Scenario

A set of options within ODI controlling some aspect of the extraction, loading and transforming of data being moved between source and target locations.

**Season Specific Planogram**

Planogram designed to be placed during a specific time period. An example of a season specific planogram would be one for winter clothing.

**Shelf Object**

An item of equipment that is placed on a parent fixture. Can also be called a Fixel.

**Source Bridge**

The Source Bridge is one of two halves of the planogram import solution. It is maintained by the customer and is used to generate the XML file that is the data source for the Target Bridge.

**Standalone Agent**

An Agent typically installed on a batch server that can invoke the import scenarios specified in the Master and Work Repositories without invoking the full ODI functionality.

**Store Specific Planogram**

Planogram designed to go into a sub-set of stores in a retail chain. For example, some planograms might contain food designed to appeal to a Hispanic population.

**Target Bridge**

The Target Bridge is one of two halves of the planogram import solution. It is maintained by Oracle and used to take the XML file containing planogram information generated by the Source Bridge and import it into the MSP database.

**UDA**

Abbreviation for User Defined Attribute.

**User Defined Attribute**

Customer specific information that can be assigned to specific objects in the MSP database. In the case of planograms, UDAs can be assigned to the planogram itself, bays within the planogram, items of planogram equipment and products associated with the planogram.

**Work Repository**

Part of a database instance used by ODI to store information about the details of a data import. A Work Repository has to be associated with a Master Repository.