

Oracle® Data Integrator

Installation Guide

10g Release 3 (10.1.3)

September 2008

Copyright © 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table Of Contents

Document Structure	1
Installation Overview	3
Component Details	3
Installation Process.....	4
Before installation	5
System Requirements	5
Java Configuration	8
Installing JDBC / JMS drivers	8
Pre-installation Tasks for Oracle Data Quality Products	9
Installation Procedure.....	11
Installing and Uninstalling Oracle Data Integrator and Oracle Data Quality Products	11
Installing Metadata Navigator	14
Installing Lightweight Designer	15
Installing the Oracle Data Integrator Public Web Services.....	18
Creating repositories	19
Oracle Data Integrator Repositories	19
Creating Repository Storage Spaces	19
Creating the Master Repository	21
Connecting to the Master Repository	21
Creating a Work Repository.....	22
Connecting to a Work Repository	23
Administering Oracle Data Quality Products.....	25
Managing Metabases and Repositories	25
Managing Loader Connections.....	29
Other Administrative Tasks.....	32
Upgrading Sunopsis to Oracle Data Integrator	35
Important Notes	35
Step 1: Preparing to Upgrade	35
Step 2: Upgrade the Installed Components.....	37
Step 3: Upgrade the Master Repository	38
Step 4: Upgrade the Work Repositories	39
Step 5: Import the New Objects.....	39
Step 6: Finalize the Upgrade	42
Appendices.....	43
Contents of the Oracle Data Integrator Installation Directory.....	43
Manual Installation of Oracle Data Integrator	44
Running an Agent as a Windows Service	45
Run Oracle Data Integrator with Windows 95/98	49
Installing the Java Agent on iSeries and AS/400	49

This guide contains instructions for installing and upgrading Oracle Data Integrator, Oracle Data Profiling and Oracle Data Quality for Oracle Data Integrator on any platform. It is intended for administrators installing or upgrading Oracle Data Integrator.

Document Structure

This document is structured as follows:

- **Chapter 1 - Installation Overview** contains general guidelines for setting up Oracle Data Integrator.
- **Chapter 2 - Before Installation** describes system requirements and actions to carry out before setting up Oracle Data Integrator.
- **Chapters 3 through 6** explain how to set up and upgrade Oracle Data Integrator components.

Installation Overview

Oracle Data Integrator delivers unique next-generation Extract Load and Transform (E-LT) technology that improves performance and reduces data integration costs, even across heterogeneous systems. Oracle Data Integrator offers the productivity of a declarative design approach, as well as the benefits of an active integration platform for seamless batch and real-time integration.

The Data Integrator package includes **Oracle Data Profiling** and **Oracle Data Quality for Data Integrator**:

- **Oracle Data Profiling** is a data investigation and quality monitoring tool. It allows business users to assess the quality of their data through metrics, to discover or infer rules based on this data, and to monitor the evolution of data quality over time.
- **Oracle Data Quality for Data Integrator** is a comprehensive award-winning data quality platform that covers even the most complex data quality needs. Its powerful rule-based engine and its robust and scalable architecture places data quality and name & address cleansing at the heart of an enterprise data integration strategy.

Oracle Data Profiling and **Oracle Data Quality for Data Integrator** are also referred to as **Oracle Data Quality Products**.

Component Details

Oracle Data Integrator

Oracle Data Integrator is made up of the components described below:

- The modular **Repository**, made up of a **Master Repository** and of one or several **Work Repositories**. These repositories can be installed on any database engine that supports ANSI ISO 89 syntax.
- The graphical interface modules (**Topology Manager**, **Designer**, **Security Manager**, **Operator**) and the execution agents (**Agent**). These are entirely built with Java components that give access to the Repository in client/server mode.
- **Metadata Navigator**, a Servlet/JSP application that enables access to the Repository through a Web interface.
- **Lightweight Designer**, a web application to view and edit objects in the repository through a Web browser.
- **Oracle Data Integrator Public Web Services** are web services that enable users to leverage Oracle Data Integrator features in a service-oriented architecture (SOA).

Oracle Data Profiling

Oracle Data Profiling is made up of the following components:

- The **Metabase Server** contains the profiling data and metadata.
- The **Scheduler Server** handles profiling jobs.
- The **Oracle Data Quality User Interface** is the client interface common to Oracle Data Profiling and Oracle Data Quality.

- The **Oracle Data Quality ODBC Adapter** is used to connect ODBC data sources. This optional component can be installed on Windows platforms only.

Oracle Data Quality for Data Integrator

Oracle Data Quality for Data Integrator uses the same components as Oracle Data Profiling and in addition includes.

- The **Data Quality Server** is the run-time server running data quality projects.

Note: Optional country-specific directories, used for Name and Address cleansing projects with Oracle Data Quality can be purchased separately from Trillium Software. To purchase these directories, please contact Trillium Software. You will be asked to provide the following OEM key: "ODQ020207"

Installation Process

Before installation

If you are upgrading from a previous version of Oracle Data Integrator refer to the topic: Upgrading Oracle Data Integrator.

Before installing Oracle Data Integrator, the following points should be taken into account:

- Check that the systems comply with system requirements. For more information, refer to the System Requirements and Java Configuration sections.
- If you wish to access data servers using Oracle Data Integrator, requirements regarding connectivity must be taken into consideration. For more information, refer to the Installing JDBC/JMS drivers section.
- If you install Oracle Data Profiling or Oracle Data Quality for Data Integrator,; perform the Pre-installation Tasks for Oracle Data Quality Products.

Installation Procedure

Follow the procedure in the Installing Oracle Data Integrator and Oracle Data Quality Products topic to install Oracle Data Integrator, **Oracle Data Profiling** and **Oracle Data Quality for Data Integrator** software components on your client and server machines

- If using Oracle Data Integrator, you need to create the repositories. Refer to the Oracle Data Integrator Repositories topic for more information.
- You can also configure the application server components, including Metadata Navigator, Lightweight Designer, and the Public Web Service.
- To use Oracle Data Profiling or Oracle Data Quality for Data Integrator, you need to create Metabases and Loader Connections.

Before installation

System Requirements

Configuration for Oracle Data Integrator

The minimal configuration recommended pour **Data Integrator** is as follows:

Oracle Data Integrator Client

In order to use Oracle Data Integrator graphical modules (Designer, Topology Manager, Security Manager, Operator), your computer must meet the following minimum system requirements.

Minimum System Requirements:

- 256 MB RAM
- 150 MB free disk space on your hard drive
- An operating system with graphical capabilities (such as Microsoft Windows, Linux, HP-UX, or Solaris) and supporting a Java Virtual Machine 1.4.
- A Java Virtual Machine 1.4. To use Web Services with Oracle Data Integrator a Java Development Kit (SDK) 1.5 is required.
- A TCP/IP network

However, in order to take advantage of all the advanced features of Oracle Data Integrator, we highly recommend the following configuration.

Recommended System Requirements:

- 512 MB RAM
- 300 MB free disk space on your hard drive
- An operating system with graphical capabilities (such as Microsoft Windows, Linux, HP-UX, or Solaris) and supporting a Java Virtual Machine 1.5.
- A Java Development Kit (SDK) 1.5.
- A TCP/IP network.

Oracle Data Integrator Agent

In order to run an Agent, your computer must meet the following minimum system requirements.

Minimum System Requirements:

- 256 MB RAM
- 150 MB free disk space on your hard drive
- An operating system supporting a Java Virtual Machine 1.3.1
- A Java Virtual Machine 1.3.1. To use Web Services with Oracle Data Integrator a Java Development Kit (SDK) 1.5 is required.
- A TCP/IP network.

However, in order to take advantage of all the advanced features of Agent, we highly recommend the following configuration.

Recommended System Requirements:

- 512 MB RAM
- 200 MB free disk space on your hard drive
- An operating system supporting a Java Virtual Machine 1.5.
- A Java Development Kit (SDK) 1.5
- A TCP/IP network.

Oracle Data Integrator Repository

The Repository can be installed on any relational database engine supporting ANSI ISO 89 syntax, such as Oracle, Microsoft SQL Server, Sybase AS Enterprise, IBM DB2 UDB, IBM DB2/400, etc.

You can also use for evaluation purposes the demo repository provided with Oracle Data Integrator. This repository is based on an Hypersonic SQL Database. and requires no particular installation.

The storage space requirements for repositories are the following:

- **Master Repository:** 30 Mb of storage space.
- **Work Repository:** 40 Mb of storage space. Note that the storage space required may vary depending on the size of your models and projects, and on the volume of execution logs you keep in this repository.

Metadata Navigator/Lightweight Designer

In order to run a Metadata Navigator or Lightweight Designer, your computer must meet the following minimum system requirements.

Minimum System Requirements:

- 256 MB RAM
- 50 MB free disk space on your hard drive
- An operating system supporting a Java Development Kit (SDK) 1.5
- J2EE 1.4 compliant application server, such as OC4J 10.1.3 and above.
- A Java Development Kit (SDK) 1.5.
- A TCP/IP network.

However, in order to take advantage of all the advanced features, we highly recommend the following configuration.

Recommended System Requirements:

- 512 MB RAM
- 100 MB free disk space on your hard drive
- An operating system supporting a Java Development Kit (SDK) 1.5
- J2EE 1.4 compliant application server, such as OC4J 10.1.3 and above.
- A Java Development Kit (SDK) 1.5.
- A TCP/IP network.

Configuration for Oracle Data Quality Products.

The minimal configuration recommended pour **Oracle Data Profiling and Oracle Data Quality for Data Integrator** is as follows:

Oracle Data Quality Client

In order to use Oracle Data Quality, your computer must meet the following minimum system requirements.

Minimum System Requirements:

- 512 MB RAM
- 200 MB free disk space on your hard drive
- Operating Systems: Microsoft Windows 2000 Professional/Server with SP4, Microsoft Windows XP Professional with SP2, Microsoft Windows Server 2003 Standard Edition, SP1 (32 bit).
- Internet Explorer v5.5 or greater.
- A 10 Mbps TCP/IP network.

However, in order to take advantage of all the advanced features of Oracle Data Quality, we highly recommend the following configuration.

Recommended System Requirements:

- 1 GB RAM
- 400 MB free disk space on your hard drive

Oracle Data Profiling and Data Quality Server

In order to use Oracle Data Quality products, your computer must meet the following minimum system requirements.

Minimum System Requirements:

This configuration is typically for 2 Oracle Data Profiling/Quality users and an overnight data quality processing load.

- 2 GB RAM
- Disk space depends on the data processes, we recommend 1,5 GB plus 2 to 5 times the size of the data sources.
- Operating Systems:
 - **UNIX:**
 - AIX 5L (v5.1, 5.2, and 5.3)
 - Solaris 8, 9 or 10
 - HP-UX 11.i V1 (PA RISC), HP-UX 11.i V2 (IA64)
 - Linux Red Hat Enterprise Linux 2.1 and above
 - Linux Suse Version 9.2 Professional and above

The **Inetd** component (**Xinetd** for Linux) must be installed on the UNIX machine.

- **Windows:**
 - Windows: Windows 2000 Server with SP 3
 - Windows Server 2003 Standard Edition with SP 1 (32 bit)
- A 100 MB TCP/IP network.

However, in order to take advantage of all the advanced features of Oracle Data Quality products, we highly recommend the following configuration.

Recommended System Requirements:

This configuration is typically for 5 Oracle Data Profiling/Quality users and a moderate daytime data quality processing load.

- 4 GB RAM (8 GB for a power server - For 10 users and a normal daytime load)
- Disk space depends on the data processes, we recommend 1,5 GB plus 2 to 5 times the size of the data sources.
- A 1 GB TCP/IP network.

Java Configuration

Oracle Data Integrator requires a Java Virtual Machine (JRE) or Software Development Kit (SDK). The Oracle Data Integration Suite setup program includes a Java Runtime Environment. If you want to use your own Java machine, you need to install it.

The installation differs depending on your operating system, but the general procedure is to:

1. Check whether the Java Virtual Machine is already present on your workstation and whether it is a version supported by Data Integrator. Open a command line prompt and type: `java -version`.
2. Install the Java environment, by referring to your operating system documentation.

Note: The Oracle Data Integrator execution agent can work with older versions of the Java Machine (1.3.1 and 1.4). It is recommended that you use the JVM 1.5.x for all Oracle Data Integrator client installations. All clients should have identical JVMs.

Note: Using Web Services with Oracle Data Integrator requires a Java Development Kit (SDK) 1.5. The SDKs are not provided in the setup programs, and should be installed separately. You must download them from the Sun web site, or contact your OS vendor.

Installing JDBC / JMS drivers

To connect to different data servers, Oracle Data Integrator uses the JDBC (Java DataBase Connectivity) standard for databases, JNDI (Java Naming and Directory Interface) and JMS (Java Message Service) for MOM (Message Oriented Middleware).

Oracle Data Integrator does not include any third-party driver licenses.

Oracle Data Integrator requires the drivers for your different technologies to be installed in the `/drivers` sub-directory if they are not already referenced in the Java classpath.

To add a new driver for Oracle Data Integrator:

Follow these steps to add a new driver for Oracle Data Integrator.

1. Copy the driver packages (.zip or .jar files) to the `/drivers` sub-directory under the Oracle Data Integrator installation folder. Oracle Data Integrator will automatically detect the driver the next time it starts and will add these packages to the classpath. This must be done on each machine executing an Oracle Data Integrator component.
2. If the driver is used to connect a repository and you wish to use Metadata Navigator to connect to this repository, then copy the driver's packages in the `/WEB-INF/lib` sub-directory of the Metadata Navigator installation folder, then restart the application server.

Refer to the driver's documentation for specific setup operations.

Drivers added to Oracle Data Integrator do not automatically appear in the list of drivers and URL in the Oracle Data Integrator GUI. The URL and drivers must be typed in manually in the appropriate fields, using the syntax provided in the drivers' documentations.

Caution: The automatic detection of the drivers does not work on certain systems, including Windows 95 and 98. For these systems, you must add the driver's file in the `ODI_ADDITIONAL_CLASSPATH` path parameter in the `odiparams` file.

Note: A list of all the JDBC drivers available is maintained on the Sun Microsystems Web site.

Pre-installation Tasks for Oracle Data Quality Products

Identify Available Ports

During installation, **Oracle Data Quality** asks for at least 2 port numbers that are accessible from the client to the server and not blocked. These port numbers are used for the two following services:

- The Oracle Data Quality Scheduler requires a port called **Scheduler Port**.
- The Oracle Data Profiling and Quality Metabase Repository requires a port called **Repository Port**.

To find available port numbers, on the server, you should:

- On UNIX, inspect the services file, located at: `/etc/services`
- On Windows, type the `netstat -an` command to identify ports that have applications listening on them.

Select two available ports and make note of them for the setup procedure.

Note: The port numbers should be greater than 1000 and not exceed 65535, and for easy reference, should be consecutive numbers. (For example, 7600 for the **Repository Port** and 7601 for the **Scheduler Port**.)

Note: If you are planning to use ODBC datasources from a Windows-based server, you need optionally to reserve a third port for the **Oracle Data Quality ODBC Adapter**.

Installation Procedure

Installing and Uninstalling Oracle Data Integrator and Oracle Data Quality Products

Installing Oracle Data Integrator and Oracle Data Quality Products

The **Oracle Data Integrator** installation program includes three products and gives you four installation options depending on the solution you want to install:

- The complete installation includes **Oracle Data Integrator**, **Oracle Data Profiling** and **Oracle Data Quality for Oracle Data Integrator**.
- **Oracle Data Integrator** is a comprehensive data integration platform that covers all data integration requirements.
- **Oracle Data Profiling** is a data investigation and quality monitoring tool.
- **Oracle Data Quality** for Oracle Data Integrator is a comprehensive award-winning data quality platform that covers even the most complex data quality needs.

To install Oracle Data Integrator and Oracle Data Quality Products:

1. Open the directory containing the Oracle Data Integrator setup.
2. Start Oracle Universal Installer as follows:

On...	Do This...
UNIX/Linux	Enter the following command at the operating system prompt: <code>./runInstaller</code>
Windows	Double-click <code>setup.bat</code> .

The Welcome screen appears.

5. Click **Next**.

The **Select Product To Install** screen appears.

6. Select the product to install.

7. Click **Next**.

The Select Installation Type screen appears.

8. Select the installation Type

- **Client:** Installs the client components for the product you have specified in step 6.
- **Server :** Installs the server components for the product you have specified in step 6.
- **Complete:** Installs both client and server components of Oracle Data Integrator and Oracle Data Quality products you have specified in step 6.

9. Click **Next**.

10. In the Home Details screen select the Oracle home name and directory path where you want to install the products.

11. Click **Next**.

If you are installing **Oracle Data Profiling** and / or **Oracle Data Quality for Oracle Data Integrator** you will be asked to configure the Metabase servers and to define for the client the Metabase you will connect to. Ignore this step if you do not install **Oracle Data Profiling** or **Oracle Data Quality for Oracle Data Integrator**.

Depending on the installation type specified in step 8 the following screens appear:

- a. The Metabase Connection screen appears if you have selected the **Client installation type**. This screen configures the client to connect to a specific Metabase and repository. Specify the Metabase connection as follows:

Hostname: Name of the host where the Oracle Data Profiling or Oracle Data Quality for Data Integrator server components are installed. If you are installing these components on the same machine, enter `localhost`.

Scheduler Port: Port for the Oracle Data Quality scheduler. Default is 7601.

Repository Port: Port for the Metabase Repository. Default is 7600.

Repository Name: Name of the Repository.

- b. The Metabase Configuration screen appears if you have selected the **Sever installation type**.

Define the Metabase configuration as follows:

Repository Port: Port for the Metabase repository. Default is 7600.

Scheduler Port: Port for the Oracle Data Quality scheduler. Default is 7601.

Administrator Name: Metabase administrator name. Default is `madmin`.

Administrator Password: Metabase administrator password.

12. Click **Next**.

The Oracle Data Quality ODBC Adapter Configuration screen appears only if you have chosen the server or complete installation type for Oracle Data Profiling. In this case define the Adapter Configuration as described below. If not, skip this step.

Set the ODBC Adapter port as follows:

ODBC Port: 7602

13. Click **Next**.

The **Summary** screen appears.

14. Click **Install**.

The Installation Progress screen appears.

The installer automatically executes each configuration assistant in sequence, displaying the progress in the Status column. No action is required on this screen.

When installation completes, the End of Installation screen appears with information for your review.

13. Click **Exit** and confirm when prompted.

14. On UNIX platforms add the following environment variables for the user who has installed Oracle Data Integrator:

- `ODI_JAVA_HOME=<ODI_HOME>/jre/1.4.2`
- `TS_QUALITY=<ODI_HOME>/oracledq/quality_server/tsq11r0s/Software`
- `LD_LIBRARY_PATH=<ODI_HOME>/oracledq/quality_server/tsq11r0s/Software/bin`

This completes the installation procedures.

Note: Oracle Data Integrator can be installed as a standalone component without running the setup program in specific cases (platform not supported by the setup program). Refer to the Manual Installation of Oracle Data Integrator and installing the Java Agent on iSeries and AS/400 topics for more details.

Uninstalling Oracle Data Integrator and Oracle Data Quality Products

To uninstall Oracle Data Integrator and Oracle Data Quality Products:

1. Open the directory containing the Oracle Data Integrator setup.
2. Start Oracle Universal Installer as follows:

On...	Do This...
UNIX/Linux	Double-click <code>setup.sh</code> or enter the following command at the operating system prompt: <code>./runInstaller</code>
Windows	Double-click <code>setup.bat</code> .

The Welcome screen appears.

5. Click **Installed Products...**
6. Select the products to remove from the Installed Products window.
7. Select the **Remove** button.

Removing the Oracle Data Quality Services on UNIX

In order to successfully uninstall the Oracle Data Quality server, you may need to remove various directories and also entries from several files on certain UNIX platforms.

1. Verify that there are no users logged on to the Metabase Server.
2. Log on to the UNIX server as Root.
3. Stop the scheduler server
 1. Change directories to `<ODI_HOME>/oracledq/metabase_server/metabase/bin`
 2. Type the command: `./scheduler -stop`
4. Delete the data quality information
 1. Change directories to `<ODI_HOME>/oracledq`
 2. Type the command: `rm -R /metabase_server`
 3. Type the command: `rm -R /metabase_data`. Note that this command deletes all the repository and metabase data.
5. Remove the services
 1. Make a backup copy of the `/etc/services` file.
 2. Open this file for editing and go to the end of the file.
 3. Here you will see two lines referring to services you have created for the metabase server and scheduler.
 4. Delete these two lines from the services file.
 5. Save your changes and close the file.

6. Remove the inetd configuration the entry corresponding to the metabase server.
 1. Make a backup copy of the `/etc/inetd.conf` file. Open this file for editing, then find in it the entry referring to the metabase serve, then remove it and save the file.
 2. If using Linux, you must remove instead the `/etc/xinetd.d/TSDiscovery` file.
7. Restart inetd.

Installing Metadata Navigator

Metadata Navigator is a Servlet/JSP application that enables access to the Repository through a Web interface. This application requires a Web server with a Servlet/JSP container such as Jakarta Tomcat or Oracle Container for Java (OC4J).

Deploying Metadata Navigator

Note: To deploy Metadata Navigator, refer to the documentation of your application server to find the recommended method for deploying a web application from a web application resource (WAR) file or from a set of JSP pages.

To install Metadata Navigator:

1. Insert the Oracle Data Integrator CD-ROM in your CD-ROM drive.
2. To install Metadata Navigator, you may either:
 - Use the web application deployment feature of the application server with the `oracledimn.war` file located in the `/setup/manual/` directory on the CD. This solution is recommended for OC4J (Use the **Deploy** button in the **Applications** tab).
 - if the application server allows it, copy the `/oracledimn` directory from the CD to the application server's web applications directory (`Webapps` for Tomcat).
3. Restart the web server hosting the application.

Install the JDBC driver

Metadata Navigator uses the JDBC connectivity to access the database server hosting the Repository to explore. It is necessary to configure the JDBC connectivity on the machine where Metadata Navigator is installed.

The Java classes of the JDBC driver must be copied in the `/WEB-INF/lib` sub-directory of the Metadata Navigator installation folder, and must have the `.jar` extension. Rename the file extension to `.jar` if necessary (e.g., from `.zip` to `.jar`). Refer to the installation documentation for the JDBC driver provided with your database for more information.

Configure Metadata Navigator Connections

Metadata Navigator is installed with default connections. It is necessary to configure the connections to your repositories in the web application.

To configure Metadata Navigator connections:

1. Configure the connection to the work repository in Designer.
2. Go to your Oracle Data Integrator installation folder's `/bin` sub-directory.

3. Copy the `snps_login_work.xml` file from this directory into `/WEB_INF` sub-directory of the Metadata Navigator deployment folder's.

Installing Lightweight Designer

Lightweight Designer is a web application to view and edit objects in the repository through a Web browser. This application uses JSF (Java Server Faces) and AJAX (Asynchronous JavaScript and XML) technologies. This application requires a Web server with a Servlet/JSP container such as Jakarta Tomcat or Oracle Container for Java OC4J.

Deploying Lightweight Designer

Note: To deploy Lightweight Designer, refer to the documentation of your application server to find the recommended method for deploying a web application from a web application resource (WAR) file or from a set of JSP pages.

To install Lightweight Designer:

1. Insert the Oracle Data Integrator CD-ROM in your CD-ROM drive.
2. To install Lightweight Designer, you may either:
 - Use the web application deployment feature of the application server with the `oracledilwd.war` file located in the `/setup/manual/` directory on the CD. This solution is recommended for OC4J (Use the **Deploy** button in the **Applications** tab).
 - if the application server allows it, copy the `/oracledilwd` directory from the CD to the application server's web applications directory (`webapps` for Tomcat), perform the configuration detailed below, then restart the web server hosting the application.

Configure the repository connections

Configuring the Datasources

Lightweight Designer connects the repositories through JDBC datasource to configure on the application server.

To configure the datasources for Tomcat:

You should configure the login for each work repository, as described below:

1. Copy the JDBC driver (.jar) files required to connect the RDBMS hosting the master and work repositories to the appropriate directory in your application server. The directory is `/common/lib` for Tomcat.
2. Define the datasources and connection information to your master and work repositories in the `META-INF/context.xml` file. If you have installed Lightweight Designer using the deployment feature of Tomcat, you should configure the datasources in the `<TOMCAT_HOME>/conf/Catalina/localhost/oracledilwd.xml` file instead. Note the name of the resource (marked in **green** below), as you will need it later. Connection information are marked in **blue**.

An example of configuration for a master and a work repository on Oracle is given below.

```
<Context >
  <Resource
    name="jdbc/ORACLE_MASTER"
    type="javax.sql.DataSource"
```

```
    auth="Container"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@SRV1:1521:ORA10"
    username="ODIM"
    password="ODIM"
    maxActive="8"
    maxIdle="2"
    maxWait="15000"
    poolPreparedStatements="true"
    removeAbandoned="true"
    initialSize="2"
    removeAbandonedTimeout="1800"
  />
  <Resource
    name="jdbc/ORACLE_WORK"
    type="javax.sql.DataSource"
    auth="Container"
    driverClassName="oracle.jdbc.OracleDriver"
    url="jdbc:oracle:thin:@SRV1:1521:ORA10"
    username="ODIW"
    password="ODIW"
    maxActive="8"
    maxIdle="2"
    maxWait="15000"
    poolPreparedStatements="true"
    removeAbandoned="true"
    initialSize="2"
    removeAbandonedTimeout="1800"
  </Context>
```

Parameters Details:

- **name** is the datasource name
- **username** and **password** in use in your database
- **driverClassName** contains the JDBC class name in the driver
- **url** contains the JDBC URL
- **maxActive** contains the maximum number of simultaneous authorized connections
- **maxIdle** contains the maximum number of simultaneous connections in idle state in the connection pool.

For more information, refer to the Apache Tomcat manual. These parameters should be adapted to the workload of your server.

To configure the datasources for OC4J:

With OC4J, you must define the datasource for each master or work repository used as follows:

1. Copy the JDBC driver (.jar) files required to connect the RDBMS hosting the master and work repositories to the appropriate directory in your application server. The directory is `ORACLE_HOME/j2ee/home/applib` for OC4J.
2. Connect to OC4J administration interface.
3. In the **Administration** tab, in **Services | JDBC Resources** click **Go to task**.
4. Click the **Create** button in the **Connection Pools** section.
5. Select the Lightweight Designer application, select **New Connection Pool** , then **Continue**.
6. Fill in the fields for the JDBC connection then click **Finish**.
7. Click the **Create** button in the **Data Sources** section.
8. Select the Lightweight application, select **Managed Datasource** , then **Continue**.
9. Fill in the fields for the JDBC datasource then click **Finish**.

For more information on configuring datasources for OC4J, refer to the application server documentation.

Configuring the Repository Connections

This configuration is common to all application servers.

1. Edit the `WEB-INF/web.xml` file in the web application deployment directory and add the entries to link the datasource in the application context. An example is given below.
To reference the datasources defined above :

```
<resource-ref>
    <description>Oracle Datasource for the Master
    Repository</description>
    <res-ref-name>jdbc/ORACLE_MASTER</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
<resource-ref>
    <description>Oracle Datasource for the Work
    Repository</description>
    <res-ref-name>jdbc/ORACLE_WORK</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

2. The datasources are configured. You should now define the Lightweight Designer login information to the work repositories. Edit the `repositories.xml` file and declare a login as below for each work repository you wish to connect:

```
<login name="My Work Repository">
    <master name="jdbc/ORACLE_MASTER"
    masterDriver="oracle.jdbc.OracleDriver" />
    <work name="jdbc/ORACLE_WORK" workName="WORKREP" />
</login>
```

You need to provide in this configuration item the datasource names for the work repositories and its related master repository. You should also provide the work repository name, and the JDBC driver used to connect the master repository.

3. Restart your application to take the configuration changes into account.

Installing the Oracle Data Integrator Public Web Services

The Oracle Data Integrator Public Web Services are web services that enable users to leverage Oracle Data Integrator features in a service-oriented architecture (SOA). It provides operations such as starting a scenario.

This component can be deployed in any web service container normally a Java application server. It is provided as a Axis2 Archive File. In the examples below, Apache Tomcat 5.5 is used as the application server, with Apache Axis2 as the Web Services container. These examples may need to be adapted if using other Web Services containers.

Note: The Oracle Data Integrator Public Web Services and the Data Services are two different type of web services. The first ones allow you to access Oracle Data Integrator features through web services, the second ones are generated by Oracle Data Integrator to give you access to your data through web services.

To install the Oracle Data Integrator Public Web Services on Axis2:

1. In Axis2, go to the **Administration** page.
2. Select the **Upload Service** link
3. Browse for the Oracle Data Integrator Web Services .aar file. It is located in the `/tools/web_services/` sub-directory in the Oracle Data Integrator installation directory.
4. Click the **Upload** button.

Axis2 uploads the Oracle Data Integrator Web Services. You can now see Data Integrator Public Web Services in the Axis2 services list.

Note: Oracle Data Integrator comes with a WSIL file allowing to consume the public web services. To use it, you have to copy `OdiInvoke.wsil` file in the folder of the deployed axis2 web application (`<webapps directory>\axis2\OdiInvoke.wsil`)

Creating repositories

Oracle Data Integrator Repositories

There are two types of repositories:

- **Master Repository:** Data structure containing information on the topology of the company's IT resources, on security and on version management of projects and data models. This repository is stored on a relational database accessible in client/server mode from the different Oracle Data Integrator modules.
In general, you need only one master repository.
- **Work Repository:** Data structure containing information on data models, projects, and their use. This repository is stored on a relational database accessible in client/server mode from the different Oracle Data Integrator modules.
Several work repositories can be designated with several master repositories if necessary. However, a work repository can be linked with only one master repository for version management purposes.

The steps to follow to create repositories are detailed below:

- Creating repository storage spaces
- Creating the master repository
- Connecting to the master repository
- Creating a work repository
- Connecting to the Work repository

Note: It is recommended to regularly perform the following maintenance operations: Purge of the execution log, in order to reduce the work repository size, and backup of the databases hosting the repositories.

Note: Oracle Data Integrator includes out of the box a repository which can be used for test and demonstration purposes, through the script `/bin/startdemo`. See Getting Started for more information on the demo environment.

Creating Repository Storage Spaces

Repositories can be installed on any database engine supporting ANSI ISO 89 syntax. In each database that is to hold a Repository, a storage space must first be created.

Note: A number of database engines have been validated in our labs as storage space for Repositories, including: Hypersonic SQL, IBM DB2 UDB, IBM DB2/400, Informix, Microsoft SQL Server, Oracle, Sybase AS Anywhere, Sybase AS Enterprise, Sybase ASIQ, etc. Contact the technical support team if the database engine you wish to use is not listed above, or to receive the updated list of supported engines.

Caution: For reasons of maintenance and back-up, we strongly recommend that repositories be stored in a different space from where your application data is kept (for example in a different SCHEMA under ORACLE, or even in a different database under SYBASE or Microsoft SQLServer).

Important Note: Version management is supported for master repositories installed on the following database engines: Hypersonic SQL, IBM DB2 UDB, IBM DB2/400, Informix, Microsoft SQL Server, Oracle, Sybase AS Anywhere, Sybase AS Enterprise.

Your master repository can be stored in the same schema as one of your work repositories. On the other hand, you cannot create two different work repositories in the same schema.

A minimum of 30 MB of space should be provided for a master Repository and a minimum of 40 MB for a work Repository. These values may be increased:

- For the master repository, if using version management, 200 MB of space is recommended.
- For the work repository, to keep the history of the execution log, 200 MB is recommended.

The examples below are supplied as a guide:

Technology	Steps to follow
Oracle	<p>Create a schema snpm to host the Master repository and a schema snpw to host the work repository.</p> <p>The schemas are created by the following SQL commands:</p> <pre>SQL> create user <MY_SCHEMA> identified by <MY_PASS> default tablespace <MY_TBS> temporary tablespace <MY_TEMP>; SQL> grant connect, resource to <MY_SCHEMA>;</pre> <p>Where:</p> <ul style="list-style-type: none"><MY_SCHEMA> corresponds to the name of the schema you want to create<MY_PASS> corresponds to the password you have given it<MY_TBS> the Oracle tablespace where the data will be stored<MY_TEMP> temporary default tablespace <p>Note: Installing the repositories on Oracle versions before 9i</p> <p>The Repository installation and the temporary objects creation use the CHAR semantics syntax for creating VARCHAR columns on Oracle. This syntax is not supported on Oracle version before 9i. To install a repository on such an Oracle version, edit the <code>TECH_Oracle.xml</code> file in the <code>/lib/scripts/xml/</code> sub-directory of Oracle Data Integrator installation folder, and replace the <code>VARCHAR2 (%L CHAR)</code> string by <code>VARCHAR2 (%L)</code>.</p>
Microsoft SQL Server or Sybase ASE	<p>Create a database db_snpm to host the Master repository and a database db_snpw to host the work repository. Create two logins snpm and snpw which have these databases by default.</p> <p>Use Enterprise Manager to create the two databases db_snpm and db_snpw (allow about 40 Mb for Data and 20 Mb for Log for each of them)</p> <p>Use Query Analyzer or I-SQL to launch the following commands:</p> <pre>CREATE LOGIN <mylogin> WITH PASSWORD = '<mypass>', DEFAULT_DATABASE = <defaultbase>, DEFAULT_LANGUAGE = us_english; USE <defaultbase>;</pre>


```
CREATE USER dbo FOR LOGIN <mylogin>;
GO
```

Where:

<mylogin> corresponds to snpm or snpw

<mypass> corresponds to a password for these logins

<defaultbase> corresponds to db_snpm and db_snpw respectively

DB2/400

Create a library **snpm** to host the Master repository and a schema **snpw** to host the work repository. Create two users **snpm** and **snpw** who have these libraries by default.

Note: the libraries must be created in the form of SQL collections

Creating the Master Repository

Creating the master repository consists of creating the tables and the automatic importing of definitions for the different technologies.

To create the master repository:

1. In the **Start Menu**, select **Programs > Oracle Data Integrator > Repository Management > Master Repository Creation**, or Launch `bin/recreate.bat` or `bin/recreate.sh`.
2. Complete the fields:
 - **Driver:** the driver used to access the technology which will host the repository. For more information, refer to the section JDBC URL Sample.
 - **URL:** The complete path for the data server to host the repository. For more information, refer to the section JDBC URL Sample.
 - **User:** The user id / login of the owner of the tables (previously created under the name **snpm**).
 - **Password:** This user's password.
 - **ID:** A specific ID for the new repository, rather than the default 0. This will affect imports and exports between repositories.
 - **Technologies:** From the list, select the technology your repository will be based on.
 - **Language:** Select the language of your master repository.
3. Validate by **OK**.

Creating the dictionary begins. You can follow the procedure on your console. To test your master repository, refer to the section Connecting to the master repository.

Connecting to the Master Repository

To connect to the Master repository:

1. In the **Start Menu**, select **Programs > Oracle Data Integrator > Topology Manager**, or launch the **Topology Manager** script (`bin/topology.bat` or `bin/topology.sh`)
2. Click on the button **New** (first button to the right of the field **Login name**)
3. Complete the fields:

Oracle Data Integrator Connection:

- **Login name:** A generic alias (for example: Repository:
- **User:** SUPERVISOR (use capitals)
- **Password:** SUNOPSIS (use capitals)

DBMS Connection (Master Repository):

- **User:** User id / login of the owner of the tables you have created for the master repository
 - **Password:** This user's password
 - **Drivers' List:** choose the driver required to connect to the DBMS supporting the master repository you have just created
 - **URL:** The complete path of the data server hosting the repository. For more information, refer to the section JDBC URL Sample
4. Click on **Test** to check the connection is working.
 5. Validate by **OK**, then **OK. Topology Manager** opens.

Important Note: The default password of the SUPERVISOR account is SUNOPSIS. For security reasons, you should change this password as soon as you can.

Creating a Work Repository

Several work repositories can be designated with several master repositories if necessary. However, a work repository can be linked with only one master repository for version management purposes.

For more information on work repositories, see the reference manual.

To launch a work repository creation:

1. Connect to your master repository through the module **Topology**. For more information, refer to the section Connecting to the master repository.
2. In the icon list **Topology -> Repositories -> Work repositories**, click with the right button, then choose **Insert work repository**. A window appears, asking you to complete the connection parameters for your work repository.
3. In the **connection** window, complete the following parameters:
 - **Name:** Type the name for your work repository connection.
 - **Technology:** Choose the technology of the server to host your work repository.
 - **User:** User id / login of the owner of the tables you are going to create and host of the work repository.
 - **Password:** This user's password.
 - Tab **JDBC -> JDBC Driver:** The driver required for the connection to the DBMS to host the work repository. For more information, refer to the section JDBC URL Sample.
 - Tab **JDBC -> URL JDBC:** The complete path of the data server to host the work repository. For more information, refer to the section JDBC URL Sample.
4. Click on **Test**.

Caution: do not attempt to close this window by **OK** if you haven't tested your connection properly.

5. Click on **OK** to validate the parameters for connecting with the server to host your work repository. A window appears, asking you to give a unique name and user id code number to your repository.
6. In the window **Work Repository**, complete the following parameters:
 - **ID**: give a unique number to your repository, from 1 to 998 included.
 - **Name**: give a unique name to your work repository (for example: WorkRep1).
 - **Type**: Choose "Designer" in the list.
7. Validate by **OK**. The creation of your work repository begins and you can follow the different steps on the console.
8. When the work repository has been created, the Work Repository window closes. You can now access this repository through the modules **Designer** and **Operator**. For more information, refer to the section Connecting to the work repository.

Connecting to a Work Repository

To connect to a work repository and launch the Designer module:

1. In the **Start Menu**, select **Programs > Oracle Data Integrator > Designer**, or launch the **Designer** script (`bin/designer.bat` or `bin/designer.sh`).
2. Click on the button **New** (first button to the right of the field **Login name**).
3. Complete the fields:
 - Oracle Data Integrator Connection**:
 - **Login name**: A generic alias (for example: Repository)
 - **User**: SUPERVISOR (in capitals)
 - **Password**: SUNOPSIS (in capitals)
 - DBMS connection (Master Repository)**:
 - **User**: User id/login of the owner of the tables you have created for the master repository (not the work repository).
 - **Password**: This user's password.
 - List of drivers: choose the driver required to connect to the DBMS hosting **the master repository** you have just created.
 - **URL**: The complete path for the data server hosting the **master repository**. For more information, refer to the section JDBC URL Sample.
 - Work Repository**:
 - **Work repository name**: The name you gave your work repository in the previous step (WorkRep1 in the example). You can display the list of work repositories available in your master repository by clicking on the button to the right of this field.
4. Click on **Test** to check that the connection is working.
5. Click **OK**. The module **Designer** opens.

Important Note: The default password of the SUPERVISOR account is SUNOPSIS. For security reasons, you should change this password as soon as you can.

Administering Oracle Data Quality Products

Managing Metabases and Repositories

This chapter describes steps for configuring and managing Metabases for Oracle Data Quality Products.

Overview

In the Oracle Data Quality products (Oracle Data Profiling and Oracle Data Quality for Data Integrator), a Repository is a collection of one or more Metabases that have their own set of users, data connections, security, and performance settings.

Each Oracle Data Profiling or Data Quality Server installation installs a single Repository. An administrator must then configure Metabases for the installed Repository. Metabase configuration is performed using the Metabase Manager.

The Metabase Manager is an administrative tool that gives you access to each Repository allowing you to manage Metabases, add Oracle Data Quality Users, create Loader Connections, and configure security.

There can be only one Repository per installation of the Oracle Data Quality Server Application. On Windows servers, you can install one Repository per server. On UNIX servers, you can install multiple Repositories on one server. The Oracle Data Quality user interface (client) accesses the Repositories you define for your site.

Using the Metabase Manager

The Metabase Manager is an administrative interface that lets the Metabase administrator add one or more Metabases to a Repository and set up the connections to data sources. It also allows an administrator to manage the users who will have access to the data in the Metabases.

You can access the Metabase Manager from the Windows Start menu.

To log on to the Metabase Manager

1. From a Oracle Data Profiling or Quality client, open the Metabase Manager by clicking on the Windows Start Menu and selecting **Programs > Oracle > Oracle Data Profiling and Quality > Metabase Manager**. After the Metabase Manager displays, select a **Repository** from the list.
2. One Repository is defined during the Oracle Data Profiling or Quality client installation.
3. Enter the **Username** and **Password** for the Metabase administrator that was defined during the server installation (`madmin`).
4. Click **OK**.

Get Started with the Metabase Wizard

A Metabase Administrator must add and configure the Metabases in a Repository and make them accessible to users. At minimum, the Metabase must be connected to specific data sources (for data import), and users must be given access to the Metabases in which they plan to work.

Additionally, security needs to be set up so that unauthorized users do not have access to the data in a Metabase.

If you are setting up a Metabase for the first time, you may find it helpful to use the Get Started Wizard in the Metabase Manager. The Get Started Wizard describes the first steps you must take to configure a Metabase, add users and loader connections, and modify settings.

To open the Get Started Wizard

1. From the Metabase Manager main menu, select **Tools > Get Started Wizard**. A step-by-step description of how to use the Wizard displays in the right pane.
2. Scroll through the text and follow the steps to configure your first Metabase.

Managing Your Repository

Most of this chapter describes how to manage the Repositories in your Oracle Data Quality. Use the Metabase Manager to perform all Metabase management tasks. In addition, routinely perform backups and security checks.

Manage Metabases

The Metabase Manager allows you to manage and configure Metabases for a given Repository.

To add a Metabase

1. Log on the Metabase Manager.
2. Expand the **Control Admin** folder.
3. Right-click **Metabases** and select **Add Metabase....**
4. Refer to the information below to fill in the fields.

Field	Description
Name	Specifies a descriptive name for the Metabase.
Default Pattern	Specifies which profiling pattern representation to show in the client interface. For more information, see About About Examining Pattern topic in the <i>Oracle Data Profiling and Data Quality for Data Integrator Help</i> .
Public Cache Size (in Megabytes)	Defines how much server memory is used by each Metabase. The larger the value, the better drill-down performance. Be sure to change this value to represent your site specific requirements. Warning: Performance will rapidly decline if total cache for all Metabases exceeds available server memory.

5. Click **OK**.

To edit a Metabase

You must be logged on the Metabase Manager to perform these steps.

1. Log on the Metabase Manager.
2. From the Explorer, expand the folder named **Control Admin**.
3. From the Explorer, double-click on the item named **Metabases**. The List View displays a list of all Metabases defined in the Repository.
4. From the List View, right-click on a Metabase name and select **Edit... .**

5. To make changes, refer to the options for creating a Metabase and click **OK**. If you change the Public Cache Size setting, you must perform the following steps to activate it. Otherwise, the setting change will not take effect.
 - a. **IMPORTANT**. Make sure that no clients are accessing the Metabase that you just edited.
 - b. From the Explorer, double-click on the folder **Metabases**.
 - c. From the List View, right-click on the Metabase where you changed the **Public Cache Size** setting and select **Recover....**

To delete a Metabase

You must be logged on to the Metabase Manager to perform these steps.

1. Log on the Metabase Manager.
2. From the Explorer, expand the folder named **Control Admin**.
3. From the Explorer, double-click on the item named **Metabases**. The List View displays a list of all Metabases defined in the Repository.
4. From the List View, right-click on a Metabase name and select **Delete....**

Manage Users

There must be at least one User defined in order to access a Metabase within a Repository. Only the Metabase Administrator can add, edit and delete Users.

To add a User

1. Log on the Metabase Manager.
2. Expand the **Control Admin**. folder (left pane).
3. Right-click **Users** and select **Add User....**
4. Refer to the information below to fill in the required fields.

Field	Description
Name	Descriptive name for the Oracle Data Quality User.
Expire Password	Check this box if you want the password to expire now. The next time this User logs on to the Oracle Data Quality, they will be prompted to specify a new password.
Password	Specify a password for the user. By default, there are no password validation checks. You can customize password validation for all Oracle Data Quality Users.
Re-type Password	Re-type the password.

5. Click OK.

You can also edit a user by double-clicking it in the list of users. and delete it using its **Delete...** context menu.

Manage Metabase Access

Before a user can log on and open a Metabase, they must be granted permissions from the Metabase Administrator.

You must be logged on the Metabase Manager as the Metabase Administrator to perform this task.

To grant Metabase access to users

1. Log on the Metabase Manager.
2. Expand the folder named **Control Admin**.
3. From the Explorer, right-click **Metabase Users** and select **Add User to Metabase....**
4. Refer to the information below to fill in the required fields.

Field	Description
Username	Select a User from the list. If you do not see a User, make sure that you created one.
Metabase	Select a Metabase from the list. If you do not see a Metabase, make sure that you created one.
Limited User	Place a check in this box to designate this User as Limited. A Limited User can only view Entities they have created or Entities that other Users have given them access to view. If a Limited User is not granted access to an Entity, they will not see any object involving that Entity such as Joins, Notes, Bookmarks, etc.

5. Click **OK**.

Adding Connections to Additional Repositories

After the Oracle Data Quality User Interface is installed, your site might decide to add another Repository to your enterprise. In order to access this new Repository, you must first establish a connection from the Oracle Data Quality User Interface.

To add a connection to a new Repository

1. Log on the Oracle Data Quality User Interface client machine as the Windows administrator
2. From the Windows **Control Panel**, open **Add or Remove Programs**.
3. Locate the entry Oracle Data Quality User Interface.
4. Click **Change**.
5. Click **Next**.
6. Click **Modify**.
7. Click **Next**.
8. At the **Metabase Repository** window, click **More** and refer to the information below to add a connection to the new Repository then click **Next**.

Field	Description
Repository Name	Local name for the Repository stored on the Oracle Data Quality Server. This name is displayed in the login screens of this client. Choose any name that is meaningful to the person accessing the Repository.
Repository Host Name	Actual server name where the Repository is stored.

Repository Port	Metabase Login Port defined during Oracle Data Quality Server application installation.
-----------------	---

9. Click **Next** until you are prompted to **Install**, then click **Install**.
10. Click **Finish** when maintenance is complete.

Managing Loader Connections

Manage Loader Connections

To import data into a Metabase, you must have at least one Loader Connection for each type of data source that you intend to profile.

You must be logged on the Metabase Manager as the Metabase Administrator to add, edit, and delete Loader Connections.

Important Note: A Loader Connection is accessible by all Metabases in a Repository; therefore, take precautions to ensure that security for data sources is adequate and will prevent unauthorized access.

To add a Loader Connection

1. Log on the Metabase Manager.
2. Expand the **Control Admin** folder.
3. Right-click Metabase **Loader Connections** and select **Add Loader Connection....** You can also select **Tools > Add Load Connection** from the Metabase Manager main menu.
4. Specify a **Name** and **Description**. You'll see this name and description again in the Create Entity Wizard of the client user interface.
5. Select a **Type**. No matter which **Type** is selected, you will have the option to specify a **Default Filter**. This filter limits the number of files to display in the **Create Entity Wizard** of the client interface. The default is *, which indicates all files in the specified directory will be shown.
6. Depending on the type that you select, you will be prompted for type-specific parameters. Fill in the type-specific parameters.

You can also edit a loader connection by double-clicking it in the list of users. and delete it using its **Delete...** context menu.

To view all defined Loader Connections

1. Log on the Metabase Manager.
2. Expand the **Control Admin** folder.
3. Double-click **Loader Connections**. A list of all Loader Connections defined for this Repository (and all its Metabases) is displayed in the List View pane.
4. Examine row information to find out the following.

Field	Description
Name	Name of the Loader Connection - displayed in the Import Data window of the client application.
Type	Type of the Loader Connection. Valid types are COBOL, DB2, Delimited, ODBC, ORACLE, and Trillium.

Description	Description of the Loader Connection - displayed in the Import Data window of the client application.
Parameters	Parameters describing file locations or relational source information. Varies depending on the Type.
Filter	Default filter used in the Create Entity Wizard .
Edited By	User that last edited the connection.
Edited Date	Date that the connection was last modified.
Created By	User that created the connection.
Created Date	Date that the connection was created.

Changing Loader Settings to Optimize Performance

There are four settings in the Metabase Manager that can impact data import performance:

- **Loader Cache Size:** maximum memory available for load process
- **Run Dependencies at Load Time:** enables Dependency Analysis
- **Run Keys at Load Time:** enables Key Analysis
- **Find Duplicate Rows at Load Time:** enables Duplicate Key discovery

These settings are available in the **Edit Loader Settings** option when you right-click **Loader Connections** in the **Control Admin**. They are described in detail in the following topics.

Change Loader Cache Size

The Loader Cache Size specifies the maximum memory allowed for the load process. If possible, configure the cache size value to be the maximum size your hardware allows without relying on virtual memory.

To change the loader cache size

Note: Changing these settings affects the entire Repository.

1. Log on the Metabase Manager.
2. Expand **Control Admin**.
3. Right-click **Loader Connections** and select **Edit Loader Settings....**
4. Make changes based on the information below.

Setting	Description
Loader Cache Size	<p>Specifies maximum memory allowed for the load process.</p> <p>**Be sure to change this value to be as large as your site's hardware will tolerate without relying on virtual memory.</p> <p>On multiple CPU systems, this process can run multiple times in parallel. In this scenario, memory is shared across all processes to ensure that it does not exceed the limit you define.</p>

5. Click **OK**.

Disable Key/Dependency Checks at Load Time

By default, Oracle Data Profiling automatically performs Key and Dependency analysis on a sample of your data (10,000 rows) during data import to find possible Keys and Dependencies. If you are not interested in finding Keys or Dependencies during data import, you can disable this setting.

Disabling the Key and/or Dependency Analysis improves the load performance when you import data. However, if the Dependency or Key Analysis is not run during the data import process, you must manually perform the analysis using the Oracle Data Quality user interface. For more information, see the Oracle Data Quality user interface Help.

To disable Key and Dependency Analysis on data import

1. Log on the Metabase Manager
2. Expand **Control Admin**.
3. Right-click **Loader Connections** and select **Edit Loader Settings...**
4. Make changes using the information below.

To disable...	Description
Dependency analysis at load time	<ol style="list-style-type: none"> 1. Click Run Dependencies at Load Time. 2. Remove the check from the box.
Key analysis at load time	<ol style="list-style-type: none"> 1. Click Run Keys at Load Time. 2. Remove the check from the box.

5. Click **OK**.

Set Duplicate Rows at Load Time

By default, Oracle Data Profiling finds duplicate rows when data is imported to a Metabase. You can improve load performance when you disable this setting.

If you do not want duplicate rows identified, follow these steps:

To disable finding duplicate rows at load time

1. Log on the Metabase Manager.
2. Expand the **Control Admin** folder.
3. Right-click **Loader Connections** and select **Edit Loader Settings....**
4. Make changes based on the information below.

To disable...	Description
duplicate rows at load time	<ol style="list-style-type: none"> 1. Click the tab Find Duplicate Rows at Load Time. 2. Remove the check from the box.

5. Click **OK**.

To reduce load time by specifying a processing limit

If the number of duplicate rows discovered has the potential to be high, you can limit the number of duplicate rows found during the initial processing.

1. Follow Steps 1 through 4 above.
2. Verify that **Discover duplicated rows** is checked.

3. In **Potential duplicates threshold**, type a number to specify a limit for the number of duplicated rows the initial process is allowed to find.
4. Click **OK**.

Other Administrative Tasks

Enable E-mail Notification

Oracle Data Quality for Data Integrator can notify Oracle Data Quality users via e-mail when the following events occur:

- Business Rules fail
- Business Rules succeed
- Exported rows complete
- Exported rows fail
- Load job completes
- Load job fails

Using the Email Settings in the Metabase Manager, you can set up an authenticated SMTP server. An authenticated SMTP server requires a username and password to send mail or to change the port on which the SMTP server listens.

After you have enabled e-mail notification, you can configure how the e-mails are constructed using the Oracle Data Quality User Interface.

To enable e-mail notification with authentication

1. Log on the **Metabase Manager**.
2. From the main menu, select **Tools > Edit Email Settings....**
3. Enter the following information:
 - Email From:** Name to display as the sender of the e-mail. Depending on how your mail server is configured, you may be required to enter a qualified e-mail address.
 - Email Server:** Host name of an SMTP server that will send the e-mail.
 - Email Login:** Username required to log on to the server.
 - Email Password:** Password required to log on to the server.
4. Click **OK**.

Manage Password Security

You must be logged on as the Metabase Administrator (madmin) to perform these tasks.

To edit password properties for all users

1. Log on the Metabase Manager.
2. Expand **Control Admin**.
3. Right-click Users and select **Edit Password Properties....**
4. Refer to the information below to make changes and click **OK**.

Field	Description
Invalidate user	Determines how long a new password is valid. At the end of this time

password after	period, the user interface prompts the user for a new password and the cycle begins again.
Number of unrepeatable old passwords:	<p>Number of times an old password can be set.</p> <p>Example 1</p> <p>Only allow passwords with a minimum of 6 characters.</p> <p>Type the regular expression: <code>. { 6 }</code></p> <p>Example 2</p> <p>Only allow passwords that are:</p> <ul style="list-style-type: none"> • 5 through 16 characters long and • contain only alpha numerics (including underscore <code>_</code> character). <p>Type the regular expression: <code>{ ^ \w { 5 , 16 } \$ }</code></p> <p>Explanation:</p> <p>Match whole string using the <code>^</code> (start) and <code>\$</code> (ending) meta-characters. Next we say we want to match alpha numerics (<code>[a - z A - Z 0 - 9]</code>) for which we use the <code>\w</code> meta-character. And we add the constraint of 5 characters minimum and 16 characters maximum <code>{ 5 , 16 }</code></p>
Valid Password regular expression:	<p>Example 3</p> <p>Only allow passwords that:</p> <ul style="list-style-type: none"> • are at least 10 characters long, • contain at least one lower case letter, one upper case letter, one digit, and one special character • allow the special characters (which are configurable): <code>@ # \$ % ^ & + =</code> <p><code>{ ^ . * (? = . { 10 , }) (? = . * \d) (? = . * [a - z]) (? = . * [A - Z]) (? = . * [@ # \$ % ^ & + =]) . * \$ }</code></p>
Invalid Password regular expression:	<p>Example 1</p> <p>If you type the regular expression <code>. { 9 }</code> then only those passwords with a maximum length of 8 characters are valid.</p> <p>Example 2</p> <p>If you type the regular expression <code>{ ^ [a - z A - Z] * \$ }</code> then passwords that contain only alphabetic characters (of upper and lower case) are invalid. This effectively means that passwords are required to contain non-alphabetic characters.</p>

Upgrading Sunopsis to Oracle Data Integrator

The procedure below details the process for upgrading from a Sunopsis version 3.0.00.00 and later to Oracle Data Integrator. Please read the entire procedure carefully before starting the upgrade.

Note: This procedure only applies to upgrade Sunopsis v3.0 and higher. If you want to upgrade from an Oracle Data Integrator 10gR3 (10.1.3.x) version, please refer to the section Upgrading Oracle Data Integrator. **Oracle Data Integrator 10.1.3** is the version following Sunopsis v4.1.

Important Notes

Before running the upgrade, please note the following:

- Throughout this procedure, <install_dir> refers to the existing installation directory of Sunopsis. <ODI_HOME> refers to installation directory for the new version.
- The upgrade of a master and work repository performs modifications in the repository's structure. This sequence of modifications cannot be undone and must be done in one continuous step. If an error occurs while upgrading the repository, you must restore the repository database backup before restarting the repository upgrade process.
- Oracle Data Integrator requires a Java virtual machine (JVM) 1.4. To use Web Services with Oracle Data Integrator a Java Development Kit (SDK) 1.4 is required. A JVM 1.4 is provided with the installation programs.
- Upgrading Metadata Navigator is not included in this procedure. Application servers' administrators should un-deploy then reinstall the Metadata Navigator application.

Step 1: Preparing to Upgrade

1.1: Stopping agents services

If you have started the Agent as a Windows Service, you should stop and uninstall this service before upgrading, then reinstall it afterwards. The original service configuration is stored in the backup directory.

1.2: Backup

It is essential that you perform a backup copy of the Sunopsis installation directory and repositories before starting the upgrade process.

1. Make a backup copy of the installation directory of Sunopsis (referred to hereafter as the backup directory).
2. Make a backup of the databases, libraries or schemas where the repositories are stored.

1.3: Preparing to Upgrade Profiles

Customized Profiles

Profiles are automatically reinitialized by the repositories upgrade process. If you have changed the default profiles, then you must make a copy of these modified profiles before proceeding with the upgrade. At the end of the process, you will have to reapply your changes to the profiles which will have been reset by the upgrade, using the duplicates as a reference.

If you have designed your own profiles, then you should update them at the end of the process to take into account the objects and methods added or deleted by the upgrade process. See below for a list of modified objects

Modified Objects

Objects which changed in Sunopsis v4.0, grouped by feature:

- Markers: **Markers**, **Marker Groups**. **Life Cycle States** are deprecated.
- Versioning: **Solutions**, **Solution Elements**, **Versions**. Note that versioning methods are attached to each object.

Objects which were added in Sunopsis v4.1, grouped by feature:

- Common Format Designer: **Action Groups**, **Actions**, **Action Lines**, **Model Folders**, **Diagrams**.

Objects which were added/modified in Data Integrator 10.1.3.0:

- The following object types are modified (new methods): datastore, model folders, model, variable, object, action group, action, action line, step report, version, procedure, interface, scenario, object, .
- New object types: **Scenario folder**, **Open Tools** (This object handles add/delete privileges on Open Tools)

Upgrading Profiles for Versions Prior to Sunopsis 3.2.03.00

If you are upgrading from a version prior to Sunopsis 3.2.03.00, make sure that you have the default profiles installed with the following **Internal ID** values (displayed on the **Version** tab of the **Profile** window):

- CONNECT: 24999
- DESIGNER: 25999
- NG DESIGNER: 28999
- METADATA ADMIN: 26999
- NG METADATA ADMIN: 29999
- TOPOLOGY ADMIN: 34999
- SECURITY ADMIN: 33999
- OPERATOR: 31999
- REPOSITORY EXPLORER: 32999
- NG REPOSITORY EXPLORER: 30999

If the profiles already have these internal ID values, then you can proceed with the upgrade. At the end of the upgrade, check that changes automatically made to the profiles to take into account the new objects and methods are appropriate for your security policy.

If at least one of your profiles has an identifier that does not match the list above:

1. Delete all existing users and profiles using Security Manager.
2. Perform the upgrade.
3. At the end of the upgrade process, re-define the security policy using the profiles imported by the upgrade process.

Step 2: Upgrade the Installed Components

The components upgrade needs to be performed on each machine where a Sunopsis module (Designer, Operator, Topology Manager, Security Manager or Agent) is installed. This upgrade can be performed either using a setup program (recommended) or manually.

Note: The setup program is typically used for upgrading the client machines and graphical modules and the manual setup is typically used to upgrade agents on servers.

Note: The setup program is available for a number of platforms. For other platforms, the manual upgrade process must be used. To retrieve the `/oracledi` directory necessary for the manual upgrade, download the Linux or Windows CD for Oracle Data Integrator from the Oracle Technology Network and use the `/oracledi` sub-directory located at the root of this CD.

Upgrade Notes

The installation program assumes that you install Oracle Data Integrator in a different directory from the old installation directory.

As a consequence:

- The connection settings (stored in `/bin/snp_login_work.xml` and `/bin/snp_login_security.xml` files) are not recovered.
- Configuration parameters stored in `/bin/snpparams.bat` (.sh) are not recovered.
- Demo repositories and files (in `/demo/`) as well as the old drivers files (in `/bin/drivers`) are not recovered.

You need to perform manually the steps to carry out this configuration information to the new installation folder. These operations are detailed below in the section 2.3: *Restoring the Configuration*.

2.1: Using the Setup Program to Upgrade

1. Make sure that you have made a backup of `<install_dir>`.
2. Run the appropriate setup program for your platform. For more information, see Installing Oracle Data Integrator in the Oracle Data Integrator Installation Guide. During the setup program, when prompted for the installation folder, select a folder different from the `<install_dir>` folder. This new installation folder is referred to in this procedure as `<ODI_HOME>`. Follow the instructions given by the setup program.
3. Restore the configuration as described in step 2.3, and then uninstall the old version of Sunopsis.

2.2: Upgrading Manually

1. Make sure that you have made a backup of `<install_dir>`. You will need it later.

2. If installing from an Oracle Data Integrator CD, copy the `/oracledi` directory contents of the CD into `<ODI_HOME>` directory. If installing from a downloaded archive file, decompress the Oracle Data Integrator archive file and copy the contents of the `/oracledi` directory into the `<ODI_HOME>` directory. If you transfer the files with FTP, use the BINARY transfer mode.
3. Set the following environment variable:
 - `ODI_JAVA_HOME`: This represents the installation directory of the Java Machine in use with Oracle Data Integrator. This directory should contain the java executable file. If this variable is not set, Oracle Data Integrator will use the default Java Machine.

2.3: Restoring the Configuration

You must perform the following steps for an installation made manually or using the setup program.

1. Manually copy into the installation directory any additional files saved in the backup directory during step 1, including drivers.
2. A new `odiparams` file replaces the old `snpparams`. With a text editor, restore the parameters in the new `odiparams` file in the `<ODI_HOME>/oracledi/bin` folder from the backup `snpparams` file. Once the `odiparams` file is up to date, you can remove the following environment variables if they exist as they are specified in the `odiparams` file.
 - `SNP_JAVA_HOME`
 - `SNP_HOME`
 - `SNP_ADDITIONAL_CLASSPATH`
3. Copy the following files from the backup directory into the `<ODI_HOME>/oracledi/bin` directory:
 - all files with the extension `.xml`
 - all files with the extension `.layout`

Step 3: Upgrade the Master Repository

Note: Upgrading the repositories on Oracle versions before 9i

The repository upgrade uses the CHAR semantics syntax for creating VARCHAR columns on Oracle. This syntax is not supported on Oracle version before 9i. To install a repository on such an Oracle version, edit the `TECH_Oracle.xml` file in the `/lib/scripts/xml/` directory, and replace the `VARCHAR2 (%L CHAR)` string by `VARCHAR2 (%L)`.

You must perform this step once for each master repository that you have.

To upgrade the master repository:

1. On Windows: From the **Start Menu**, select **Programs > Oracle Data Integrator > Repository Management > Master Repository Upgrade**, or run `<ODI_HOME>/oracledi/bin/mupgrade.bat`

On UNIX: Run `<ODI_HOME>/oracledi/bin/mupgrade.sh`.

The Master Repository Upgrade tool appears.

2. Select the **Login Name** corresponding to your master repository connection, or fill in the fields:
 - **Driver**: the driver used to access the technology which hosts the repository.
 - **URL**: The complete path for the data server hosting the repository.

- **User:** The id, or login, of the user who owns the tables
 - **Password:** This user's password.
 - **Technologies:** Select the technology your repository will be based on from the list.
3. Click **OK**.

The repository upgrade process will begin. You can follow its progress on the console. A message appears when the master repository has been upgraded.

Step 4: Upgrade the Work Repositories

You must perform this step once for each work repository.

To upgrade a work repository:

1. Connect to your master repository through **Topology Manager**.
2. In the **Repositories** tree view select **Work repositories**, right click on the repository you want to upgrade and choose **Upgrade**.
3. Oracle Data Integrator will now upgrade this work repository. A window appears when the work repository has been upgraded.

Repeat these steps for each work repository.

Step 5: Import the New Objects

5.1: Objects, Methods and Profiles

This Oracle Data Integrator version introduces new objects and methods, and new privileges on these objects and methods. The objects, methods are automatically added. The default profiles built into Oracle Data Integrator are automatically updated by the upgrade process to include the appropriate privileges on all new objects and methods.

5.2: Technologies

This version of Oracle Data Integrator includes new technology definitions. You must now import these new technologies.

Note the following informations:

- If you want to use the Common Format Designer feature for a given technology, you should re-import the definition for this technology. Re-importing this technology adds the Action Groups defined for the technology, and sets the **Specific Queries** required for the DDL generation.
- It is also recommended that users upgrading from a version 3.2.03.01 or older update all their technologies to benefit from the Current Date query enabling connection recovery.
- In Oracle Data Integrator 10g, most technologies have been updated with minor fixes/updates. It is recommended to updated the technologies that you use to benefit for all these changes.

New Technologies

Technologies new to Sunopsis v3.1:

- Oracle Data Integrator Engine

- XML

Technologies new to Sunopsis v3.2:

- Generic SQL
- Java Beanshell
- Javascript
- Jython
- NetRexx
- PostgreSQL
- SAP Java Connector

Technologies new to Sunopsis v4.0:

- Teradata
- Netezza
- Hyperion Essbase

Technologies new to Sunopsis v4.1:

- All technologies have been updated to support Common Format Designer.

Technologies new to Oracle Data Integrator 10.1.3:

- Axis2
- Derby
- SAS
- Salesforce.com
- Oracle BAM
- All technologies have been updated for use in Common Format Designer.

Importing Technologies

Warning! Importing a technology erases all the changes performed in the technology. It is recommended to perform copies of the technologies you have altered, and reapply the changes to the imported technologies.

1. Run **Topology Manager**. Select the **Physical Architecture** view.
2. Expand the **Technologies** node.
3. Right-click the **Technologies** node, and select **Import Technology**.
4. Select any missing or updated technologies from the `../impexp/` directory.
5. Select the **Synonymy Insert-Update** import mode.
6. Click **OK** to import the new technologies.

5.3: Languages

This Oracle Data Integrator version includes new languages. These languages are automatically updated.

Warning! Importing a language erases all the changes made to this language. It is recommended that you make a copy of the modified languages, and reapply your changes to the updated languages.

New Languages

The updated language files are in the <ODI_HOME>/oracledi/scripts/xml/ directory.

Languages new to Sunopsis v3.1:

- SQL
- SQL_FILE

Languages new to Sunopsis v3.2:

- JYTHON
- SAP

Languages new to Sunopsis v4.0 and 4.1:

- SQL (Minor Updates in 4.1)

Languages new to Oracle Data Integrator 10.1.3:

- N/A

5.4: Generic Actions

Oracle Data Integrator version includes new and updated actions enabling DDL script generation for the Common Format Designer. These actions must be imported into the master repository.

1. Run **Topology Manager**. Select the **Generic Actions** view.
2. Right-click then select **Import Action**.
3. Select all the actions from the <ODI_HOME>/oracledi/scripts/xml/ directory.
4. Select the **Synonymy Insert-Update** import mode.
5. Click **OK** to import the actions.

5.5: Knowledge Modules

New versions of existing knowledge modules and new knowledge modules are included in this version.

Although updating knowledge modules for existing projects is not required, it is recommended that you use the new knowledge modules for new projects. Existing scenarios compiled with old knowledge module should still work normally after the upgrade without having to re-generate them. However, one exception to this rule is the **ISO SQL Incremental Update IKM** (see below).

Before importing or updating knowledge modules, you should be aware of the following:

- New knowledge modules should be imported into projects in **Duplication** mode.
- Knowledge modules for new technologies should be imported **after** the associated technologies have been imported.
- Updating an existing knowledge module used into a project should be made using the **Import Replace** option in the context menu of the knowledge module to replace. This option automatically updates any interfaces using the selected knowledge modules. After replacing a knowledge module, the affected interfaces should be thoroughly checked and tested.
- If you have customized the default knowledge modules included in Oracle Data Integrator to suit your needs, you will have to manually implement the latest changes to these original knowledge modules. The details of these changes are given below.

Important exception: KIM ISO SQL Incremental Update

If upgrading from a version prior to Sunopsis v3.2.03.15, you must update this knowledge module. Any projects using this KM should be updated to use the "IKM SQL Incremental Update" provided in this release as described above. Scenarios that use this knowledge module should then be regenerated.

Details of Changes

In this knowledge module, the following change was made in the *Update Existing Rows* command. The column metadata was modified from:

```
$$CSV_COL_LST <%=odiRef.getColList("", "[COL_NAME]", "", "", "",  
"(UPD)")%> $$CSV_COL_LST_END
```

to:

```
$$CSV_COL_LST <%=odiRef.getColList("", "[COL_NAME]", "", "", "", "", "(UPD  
AND (NOT UK) AND (NOT TRG))")%><%=odiRef.getColList("", "[COL_NAME]", "",  
"", "", "(UK)")%> $$CSV_COL_LST_END
```

This change was made in order to comply with the new `$$CSV_COL_LST` metadata syntax implemented in Sunopsis v3.2.03.10. Any customized knowledge module using the old syntax should also be updated.

Step 6: Finalize the Upgrade

Reinstall the agent services which were stopped in step 1.1.

You can now run interfaces, packages, etc developed with previous versions of Oracle Data Integrator.

Your version of Oracle Data Integrator is now up to date!

Appendices

Contents of the Oracle Data Integrator Installation Directory

The Oracle Data Integrator installation directory contains the following sub-directories:

Directory	Description
/bin	Contains all the scripts for launching the different Oracle Data Integrator modules and tools. These scripts are given in detail below.
/demo	Contains all the data sets required for the demonstration supplied with Oracle Data Integrator. It also contains the demo repository.
/doc	Contains the documentation in electronic format.
/drivers	Contains all the drivers supplied with Oracle Data Integrator.
/impexp	Contains all the objects that you can import to your repository (Knowledge modules, technologies, etc).
/lib	Contains all the java libraries used by Oracle Data Integrator
/tools	Contains the external tools provided with Oracle Data Integrator

Scripts & Tools

The following table lists the scripts and tools provided in the `/bin` directory. The extension for these scripts is `.bat` for Windows scripts and `.sh` for Unix scripts.

To launch one of these scripts, go to the Oracle Data Integrator `/bin` folder and double-click on the icon of the script file, or from a command line, enter the name of the module to launch. Type `<script_name> -help` from the command line for the on-line help.

File	Description
agent	Starts an Agent.
agentscheduler	Starts a Scheduler Agent.
agentservice	Installs or removes the Agent or Agent scheduler as a Service. (Windows Only)
agentstop	Stops an Agent.
agentweb	Starts a Web Agent.
designer	Starts Designer.

<code>jython</code>	Starts a Jython Console.
<code>mimport</code>	Starts the Master Repository Import wizard.
<code>mupgrade</code>	Starts the Master Repository Upgrade wizard.
<code>operator</code>	Starts Operator.
<code>repcreate</code>	Starts the Master Repository Creation wizard.
<code>restartsession</code>	Resumes a session.
<code>security</code>	Starts Security Manager.
<code>setpath</code>	Utility script. Adds the file name passed as a parameter to the classpath. (Windows Only)
<code>odiparams</code>	Configuration script. This script contains the parameters for starting Oracle Data Integrator modules. These variables can be manually updated in the file. The description of each of the variables is contained in the script.
<code>startcmd</code>	Starts an Oracle Data Integrator command.
<code>startdemo</code>	Starts the demonstration environment.
<code>startscen</code>	Starts a the execution of a scenario.
<code>stopdemo</code>	Stops the demonstration environment.
<code>topology</code>	Starts Topology Manager.

Manual Installation of Oracle Data Integrator

Oracle Data Integrator standalone can be installed manually. This procedure can be used for operating systems for which the setup method is not available.

To manually install Oracle Data Integrator:

1. Install a Java environment.
2. If installing from an Oracle Data Integrator CD, copy the `/oracledi` directory of the CD to the installation directory. If installing from a downloaded file, uncompress the Oracle Data Integrator archive file to the installation directory.
3. Set the following environment variables:
 - `ODI_JAVA_HOME`: This represents the installation directory of the Java Machine in use with Oracle Data Integrator. This directory should contain the `/bin` sub-directory. If this variable is not set, Oracle Data Integrator will use the default Java Machine.

Caution: If you use Microsoft Windows 95 or 98, then you must perform the operations in Run Oracle Data Integrator with Windows 95/98 before proceeding.

If the installation is completed, you can proceed with repository creation, or start the Oracle Data Integrator modules to connect existing repositories.

Running an Agent as a Windows Service

The Agent or Scheduler Agent can be installed on a windows NT, 2000,2003 or XP platform as a service.

When running the setup, you must choose the **Tools** option in the setup. This option installs the Java Wrapper tool that is used to run the agent as a service.

To install the Agent as a Service:

1. Edit the `/bin/odiparams.bat` file to match your configuration. The service created uses the agent startup parameters specified in this file.
2. Launch a Windows **Command Prompt** (Windows).
3. Go to the `/bin` sub-directory of your Oracle Data Integrator installation folder.
4. Run the `agentservice.bat` script with appropriate parameters to install the service.
5. Once the service is installed, you must go to the Windows' service manager to start the service.

Syntax

```
agentservice.bat -i|-r -a|-s [<agent_name> [<agent_port>
[<wrapper_configuration_file>]]]
```

Parameters	Description
<code>-i -r</code>	<p><code>-i</code> option installs the service</p> <p><code>-r</code> option un-installs the service.</p>
<code>-a -s</code>	<p><code>-a</code> Installs the listener agent (it uses the <code>snpsagent.conf</code> parameter file)</p> <p><code>-s</code> Installs the Scheduler agent (it uses the <code>snpsagentscheduler.conf</code> parameter file)</p>
<code><agent_name></code>	<p>Name of the physical agent as declared in the topology. This parameter is mandatory for a Scheduler Agent, and optional for a listener agent.</p> <p>Note: This parameter affects the definition of the created service. When un-installing a service, you must specify the same <code><agent_name></code> that was used to install it.</p>
<code><agent_port></code>	Listening port for the agent.
<code><wrapper_configuration_file></code>	Name of the wrapper configuration file. This file must be located in the <code>tools/wrapper/conf/</code> sub-directory of your Oracle Data Integrator installation folder. This parameter is optional.

Configuration File

The agent startup configuration (connected repository, java parameters) is defined in the `odiparams.bat` file.

The wrapper configuration file, located in the `tools/wrapper/conf/` sub-directory of your Oracle Data Integrator installation folder, defines how the service will be launched. It contains part of the parameters of the agent, and may be changed according to your configuration.

For example, the following elements could be changed:

- Additional drivers need to be added in the classpath, as `wrapper.java.classpath.xx` properties.
- The properties related to the of log files management (`wrapper.logfile.xx`) may be changed depending on the log size and level required.
- Increase the ping timeout (`wrapper.ping.timeout`) for a machine with a high CPU load.

The reference for this configuration file is given at the following URL:
<http://wrapper.tanukisoftware.org/>.

Warning: You should not change a configuration file used by an existing service. The service should be un-installed first.

Configuration file sample

This sample file highlights the key parameters to adapt to your configuration, in **green**.

```
# *****
# Java wrapper Properties properties file for an Agent Scheduler
# Note that most parameters are set in the odiparams.bat file, or on
# the agent service.bat command line.
# They are displayed in this file for information.
# *****
# *** Set the path to reference the jvm used for your Agent
# This parameter is read in odiparams.bat
# wrapper.java.command=java.exe
wrapper.working.dir=../../bin/
# Java Main class.
wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp
# Java Classpath. Add class path elements as needed starting from 1
wrapper.java.classpath.1=../tools/wrapper/lib/wrapper.jar
wrapper.java.classpath.2=../lib/.
wrapper.java.classpath.3=../lib/sunopsis.zip
wrapper.java.classpath.4=../lib/snpshelp.zip
wrapper.java.classpath.5=../lib/jakarta-ant-1.4.1-optional.jar
wrapper.java.classpath.6=../lib/NetComponents.jar
wrapper.java.classpath.7=../lib/sunjce_provider.jar
wrapper.java.classpath.8=../lib/jce1_2_2.jar
wrapper.java.classpath.9=../lib/local_policy.jar
wrapper.java.classpath.10=../lib/US_export_policy.jar
wrapper.java.classpath.11=../lib/scripting/.
wrapper.java.classpath.12=../lib/scripting/bsf.jar
```

```

wrapper.java.classpath.13=../lib/scripting/bsh-1.2b7.jar
wrapper.java.classpath.14=../lib/scripting/js.jar
wrapper.java.classpath.15=../lib/scripting/jython.jar
wrapper.java.classpath.16=../drivers/.
wrapper.java.classpath.17=../drivers/classes12.zip
wrapper.java.classpath.18=../drivers/db2java.zip
wrapper.java.classpath.19=../drivers/jt400.zip
wrapper.java.classpath.20=../drivers/mysql.zip
wrapper.java.classpath.21=../drivers/snpsmq.zip
wrapper.java.classpath.22=../drivers/snpsxmlo.zip
wrapper.java.classpath.23=../drivers/sqlj.zip
wrapper.java.classpath.24=../drivers/common.jar
wrapper.java.classpath.25=../drivers/db2fs.jar
wrapper.java.classpath.26=../drivers/db2jcc.jar
wrapper.java.classpath.27=../drivers/jconn2.jar
wrapper.java.classpath.28=../drivers/xerces.jar
# *** add here the additionnal driver classes that do not appear in the
list above
# Java Library Path (location of Wrapper.DLL or libwrapper.so)
wrapper.java.library.path.1=../tools/wrapper/lib/
# Java Additional Parameters
wrapper.java.additional.1=-Djava.security.policy=server.policy
# Initial Java Heap Size (in MB)
# This parameter is read in odiparams.bat
# wrapper.java.initmemory=64
# Maximum Java Heap Size (in MB)
# This parameter is read in odiparams.bat
# wrapper.java.maxmemory=128
# Application parameters.
wrapper.app.parameter.1=oracle.odi.Agent
# The parameters below, related to the repository connection, are read
in odiparams.bat
# update them in odiparams.sh
# wrapper.app.parameter.1=oracle.odi.Agent
# wrapper.app.parameter.2="-SECU_DRIVER=org.hsqldb.jdbcDriver"
# wrapper.app.parameter.3="-SECU_URL=jdbc:hsqldb:hsqldb://localhost"
# wrapper.app.parameter.4="-SECU_USER=sa"
# wrapper.app.parameter.5="-SECU_PASS="
# wrapper.app.parameter.6="-WORK_REPOSITORY=WORKREP2"
#
# Add additionnal parameters below, starting from parameter number 10
#*****
# Wrapper Logging Properties
#*****

```

```
# Format of output for the console.  (See docs for formats)
wrapper.console.format=PM
# Log Level for console output.  (See docs for log levels)
wrapper.console.loglevel=INFO
# Log file to use for wrapper output logging.
wrapper.logfile=agentservice.log
# Format of output for the log file.  (See docs for formats)
wrapper.logfile.format=LPTM
# Log Level for log file output.  (See docs for log levels)
wrapper.logfile.loglevel=INFO
# Maximum size that the log file will be allowed to grow to before
# the log is rolled. Size is specified in bytes. The default value
# of 0, disables log rolling. May abbreviate with the 'k' (kb) or
# 'm' (mb) suffix. For example: 10m = 10 megabytes.
wrapper.logfile.maxsize=1m
# Maximum number of rolled log files which will be allowed before old
# files are deleted. The default value of 0 implies no limit.
wrapper.logfile.maxfiles=10
# Log Level for sys/event log output.  (See docs for log levels)
wrapper.syslog.loglevel=FATAL
#*****
# Wrapper NT Service Properties
#*****
# WARNING - Do not modify any of these properties when an application
# using this configuration file has been installed as a service.
# Please uninstall the service before modifying this section. The
# service can then be reinstalled.
# Name of the service
# This parameter is set depending on the agent Name
wrapper.ntservice.name=SnpsAgent
# Display name of the service
wrapper.ntservice.displayname=Oracle Data Integrator Agent
# Description of the service
wrapper.ntservice.description=Executes Oracle Data Integrator Sessions
# Service dependencies. Add dependencies as needed starting from 1
wrapper.ntservice.dependency.1=
# Mode in which the service is installed. AUTO_START or DEMAND_START
wrapper.ntservice.starttype=AUTO_START
# Allow the service to interact with the desktop.
wrapper.ntservice.interactive=false
#*****
# Wrapper JVM Timeouts in seconds (0 = No timeout limit)
#*****
```

```
# wrapper.ping.timeout=10
# wrapper.cpu.timeout=30
```

Run Oracle Data Integrator with Windows 95/98

In order to run Oracle Data Integrator on Windows 95/98, you must increase the default memory allocated to the Oracle Data Integrator application.

Increase the default memory allocated to the Oracle Data Integrator application

1. Create a shortcut for each of these files. To create a shortcut, right-click the file name and select **create shortcut** or change the shortcuts created by Oracle Data Integrator' setup:
 - designer.bat
 - topology.bat
 - startdemo.bat
2. On each shortcut, do a right-click and select **Properties**
3. Go to the **Memory** tab and
 - set the **Conventional Memory - Total** to 640 (or more if available)
 - set the **Conventional Memory - Initial Environment** to 4096 (or more if available)
 - set the **Extended (XMS) Memory - Total** to 16384 (or more if available)
 - set the **MS-DOS protected-mode (DPMI) memory** to 16384 (or more if available)
4. Click on **Apply**
5. Close the **property** window.

Afterwards, only use these shortcuts to run these components.

Installing the Java Agent on iSeries and AS/400

Some databases are installed on iSeries and AS400 machines. It is possible to install Oracle Data Integrator execution agent on these machines, in order to:

- execute the loading processes on AS/400,
- execute OS400 system commands,
- reduce the network flow if both sources and targets are on AS/400.

The installation procedure below is designed to implement an Oracle Data Integrator execution agent on an AS/400 machine.

Preparing the System

Technical Pre-requisites

- AS/400 AS/400 V5R1, or V4R4M0, with the following PTF:
 - SF61800
 - SF55849
 - SF54922

- Programs to install:
 - IBM Toolbox for Java
 - Java Virtual Machine 1.3.1
- TCP/IP Services configured and launched on the AS/400 machine

Components

The Java/JDBC components used are:

- The Oracle Data Integrator Execution Agent
- The Java Virtual Machine 1.3.1
- The JDBC drivers for database access

Note about drivers: To connect DB2/400 from an AS/400, you should consider using Driver Wrapper for AS/400. See Creating a DB2/400 Data Server for more information.

Installing the Execution Agent

Installing the files

1. Create a directory tree on the AS/400 to store the Oracle Data Integrator Files:

```
MKDIR DIR('/odi')
MKDIR DIR('/odi/bin')
MKDIR DIR('/odi/lib')
MKDIR DIR('/odi/lib/scripting')
MKDIR DIR('/odi/drivers')
```
2. Transfer the files (using FTP, for instance) from the Oracle Data Integrator CD's `/oracledi` directory to the AS/400 directories created at step 1.
The files to copy are in:

```
/oracledi/bin/
/oracledi/lib/
/oracledi/lib/scripting (if you use Oracle Data Integrator Scripting)
/oracledi/drivers (only the drivers used need to be copied)
```

Creating the Java programs

It is necessary for performances reasons, on AS/400, to transform the Java packages (`.class`, `.jar` or `.zip` files) into **Java programs**.

To create a Java program from a `.class`, `.jar` or `.zip` file:

1. Launch the CL command:

```
CRTJVAPGM CLSF('<.class, .zip or .jar file location>')
OPTIMIZE(40)
```

For instance, to create a Java program from the `odi.zip` file (the Java agent), run:

```
CRTJVAPGM CLSF('/odi/lib/odi.zip') OPTIMIZE(40)
```

Note : Creating a Java program is a long operation, depending on the nature of the Java package. It is recommended you create programs only for the classes effectively used.

Launching the Execution Agent

You can launch the agent by two means:

- **Using the Shell interpreter: (QSH or STRQSH)** It is an OS/400 option implementing a UNIX-compatible shell. A number of UNIX commands are accessible (ls, chmod, chown, etc...) as well as the 'java' command. You may launch Oracle Data Integrator using the standard syntax for the from the Unix scripts (.sh extension) provided with Oracle Data Integrator in /bin. You must configure the odiparams.sh file before running the agent. The graphical modules (Topology, Designer, ...) are usable on AS/400.
- **Using OS/400 Commands (CL):** The RUNJVA or JAVA CL commands execute a Java application. It is convenient to use a CL program in order to launch the agent or the scenarios. You will find program template for these operations below.

Note about JVM version: When multiple Java machines are installed on the AS/400, it may be necessary to force the version of Java used in the Java command.

- For **QSH**: The flag -Djava.version=<java version> (example -Djava.version=1.3.1) of the Java command is used to force the version.
 - For the **OS/400 commands**: You pass the property as a JAVA command parameter: -PROP((<property> <value>) (<property> <value>)). For instance -PROP((java.version 1.3.1)

Note about Agent Naming: On AS/400, it is frequently required to explicitly name the agent when running it as a listener. You must therefore use the agent flag -name=<agent name> when running the agent.

Running the agent

```

                PGM                PARM(&NAME &PORT &VERB)

/* Command AGENT */
/* Parameters: */
/*   &NAME: physical name of the agent */
/*   &PORT: port number */
/*   &VERB: verbose mode -V=[1..5] */
/* Example of call: */
/*   CALL PGM(<myLib/myPGM>) PARM('-NAME=myAgt' '-PORT=20910' '-V=5') */
                DCL                VAR(&NAME) TYPE(*CHAR) LEN(128)
                DCL                VAR(&PORT) TYPE(*CHAR) LEN(30)
                DCL                VAR(&VERB) TYPE(*CHAR) LEN(30)

/* All classes below should be compiled */
/* with the CRTJVAPGM command */
/* with optimize 40. */
                DCL                VAR(&PROJ) TYPE(*CHAR) LEN(512) +
                                VALUE('/odi/lib/odi.zip:+
                                /odi/lib/sunjce_provider.jar:+
                                /odi/lib/commons-net.jar:+

```

```

                                /odi/lib/local_policy.jar:+
                                /odi/lib/jakarta-ant-optional.jar:+
                                /odi/lib/US_export_policy.jar:+
                                /odi/lib/jce1_2_2.jar')
/* Replace the drivers below with your own drivers. */
        DCL          VAR(&JDBC) TYPE(*CHAR) LEN(512) +
                                VALUE('/odi/drivers/jt400Native.jar:+
                                /odi/drivers/snpsdb2.jar:+
                                /odi/drivers/ojdbc14.jar')
/* Build the Java CLASSPATH */
        DCL          VAR(&PATH) TYPE(*CHAR) LEN(1024)
        CHGVAR       &PATH (&PROJ *tcat ':' *tcat &JDBC)
/* Start the Agent */
        SBMJOB       CMD(JAVA CLASS(oracle.odi.Agent)
CLASSPATH(&PATH) +
                                PARM('&NAME &PORT &VERB') +
                                OPTIMIZE(40) +
                                OUTPUT(*PRINT))

        ENDPGM

```

Running a scenario

```

        PGM          PARM(&SCEN &VERS &CTX &VERB)
/* Command STARTSCEN */
/* Parameters: */
/*   &SCEN: scenario name */
/*   &VERS: scenario version */
/*   &CTX: context */
/*   &VERB: verbose mode -V=[1..5] */
/* Example of call: */
/*   CALL PGM(<myLib/myPGM>) PARM('myScen' 'myVers' 'GLOBAL' '-V=5') */
        DCL          VAR(&SCEN) TYPE(*CHAR) LEN(30)
        DCL          VAR(&VERS) TYPE(*CHAR) LEN(30)
        DCL          VAR(&CTX) TYPE(*CHAR) LEN(30)
        DCL          VAR(&VERB) TYPE(*CHAR) LEN(30)
/* All classes below should be compiled */
/* with the CRTJVAPGM command */
/* with optimize 40. */
        DCL          VAR(&PROJ) TYPE(*CHAR) LEN(512) +
                                VALUE('/odi/lib/odi.zip:+
                                /odi/lib/sunjce_provider.jar:+
                                /odi/lib/commons-net.jar:+
                                /odi/lib/local_policy.jar:+
                                /odi/lib/jakarta-ant-optional.jar:+

```



```

        /odi/lib/US_export_policy.jar:+
        /odi/lib/jcel_2_2.jar')
/* Replace the drivers below with your own drivers. */
        DCL          VAR(&JDBC) TYPE(*CHAR) LEN(512) +
        VALUE('/odi/drivers/jt400Native.jar:+
        /odi/drivers/snpsdb2.jar:+
        /odi/drivers/ojdbc14.jar')
/* Adapt all parameters below to your environment before use. */
        DCL          VAR(&DRV) TYPE(*CHAR) LEN(128) +
        VALUE('-
SECU_DRIVER=com.ibm.as400.access.AS400JDBCdriver')
        DCL          VAR(&URL) TYPE(*CHAR) LEN(128) +
        VALUE('-
SECU_URL=jdbc:as400://195.10.10.13;libraries=ODI')
        DCL          VAR(&USER) TYPE(*CHAR) LEN(30) +
        VALUE('-SECU_USER=QSECOFR')
        DCL          VAR(&PASS) TYPE(*CHAR) LEN(128) +
        VALUE('-SECU_PASS=XYZ')
        DCL          VAR(&WREP) TYPE(*CHAR) LEN(30) +
        VALUE('-WORK_REPOSITORY=WORKREP1')
        DCL          VAR(&SUSER) TYPE(*CHAR) LEN(30) +
        VALUE('-ODI_USER=SUPERVISOR')
        DCL          VAR(&SPASS) TYPE(*CHAR) LEN(128) +
        VALUE('-ODI_PASS=XYZ')
        DCL          VAR(&PATH) TYPE(*CHAR) LEN(1024)
/* Build the Java CLASSPATH */
        DCL          VAR(&PATH) TYPE(*CHAR) LEN(1024)
        CHGVAR        &PATH (&PROJ *tcat ':' *tcat &JDBC)
/* Execute the Scenario */
        SBMJOB        CMD(JAVA CLASS(oracle.odi.Agent)
CLASSPATH(&PATH) +
        PARM(&DRV &URL &USER &PASS &WREP &SUSER &SPASS
+
        SCEN &SCEN &VERS &CTX &VERB))
        ENDPGM

```

Note: the passwords specified must be encrypted using the command `agent ENCODE <password>`.