|  |  |
| --- | --- |
| 1) | What is Oracle Data Integrator (ODI)? |
| 2) | What is E-LT? |
| 3) | What components make up Oracle Data Integrator? |
| 4) | What is Oracle Data Integration Suite? |
| 5) | What systems can ODI extract and load data into? |
| 6) | What are Knowledge Modules? |
| 7) | How do 'Contexts' work in ODI? |
| 8) | Does my ODI infrastructure require an Oracle database? |
| 9) | Where can I get more information on ODI? |
| 10) | Does ODI support web services? |
| 11) | Where does ODI sit with my existing OWB implementation(s)? |
| 12) | Is ODI Used by Oracle in their products? |

**ANSWERS  
  
1) What is Oracle Data Integrator (ODI)?**  
Oracle acquired Sunopsis in 2006 and with it "Sunopsis Data Integrator".  
  
Oracle Data Integrator (ODI) is an E-LT (Extract, Load and Transform) tool used for high-speed data movement between disparate systems.  
  
The latest version, Oracle Data Integrator Enterprise Edition (ODI-EE) brings together "Oracle Data Integrator" and "Oracle Warehouse Builder" as separate components of a single product with a single licence.  
  
  
**2) What is E-LT?**E-LT is an innovative approach to extracting, loading and transforming data. Typically ETL application vendors have relied on costly heavy weight; mid-tier server to perform the transformations required when moving large volumes of data around the enterprise.   
  
ODI delivers unique next-generation, Extract Load and Transform (E-LT) technology that improves performance and reduces data integration costs, even across heterogeneous systems by pushing the processing required down to the typically large and powerful database servers already in place within the enterprise.  
  
  
**3) What components make up Oracle Data Integrator?**"Oracle Data Integrator" comprises of:  
  
- Oracle Data Integrator + Topology Manager + Designer + Operator + Agent  
- Oracle Data Quality for Data Integrator  
- Oracle Data Profiling

**4) What is Oracle Data Integration Suite?**

Oracle data integration suite is a set of data management applications for building, deploying, and managing enterprise data integration solutions:

* Oracle Data Integrator Enterprise Edition
* Oracle Data Relationship Management
* Oracle Service Bus (limited use)
* Oracle BPEL (limited use)
* Oracle WebLogic Server (limited use)

Additional product options are:

* Oracle Goldengate
* Oracle Data Quality for Oracle Data Integrator (Trillium-based DQ)
* Oracle Data Profiling (Trillium based Data Profiling)
* ODSI (the former Aqualogic Data Services Platform)

**5) What systems can ODI extract and load data into?**  
ODI brings true heterogeneous connectivity out-of-the-box, it can connect natively to Oracle, Sybase, MS SQL Server, MySQL, LDAP, DB2, PostgreSQL, Netezza.  
  
It can also connect to any data source supporting JDBC, its possible even to use the Oracle BI Server as a data source using the jdbc driver that ships with BI Publisher

**6) What are Knowledge Modules?**  
Knowledge Modules form the basis of 'plug-ins' that allow ODI to generate the relevant execution code , across technologies , to perform tasks in one of six areas, the six types of knowledge module consist of:

* Reverse-engineering knowledge modules are used for reading the table and other object metadata from source databases
* Journalizing knowledge modules record the new and changed data within either a single table or view or a consistent set of tables or views
* Loading knowledge modules are used for efficient extraction of data from source databases for loading into a staging area (database-specific bulk unload utilities can be used where available)
* Check knowledge modules are used for detecting errors in source data
* Integration knowledge modules are used for efficiently transforming data from staging area to the target tables, generating the optimized native SQL for the given database
* Service knowledge modules provide the ability to expose data as Web services

ODI ships with many knowledge modules out of the box, these are also extendable, they can modified within the ODI Designer module.

**7) How do 'Contexts' work in ODI?**  
ODI offers a unique design approach through use of Contexts and Logical schemas. Imagine a development team, within the ODI Topology manager a senior developer can define the system architecture, connections, databases, data servers (tables etc) and so forth.   
  
These objects are linked through contexts to 'logical' architecture objects that are then used by other developers to simply create interfaces using these logical objects, at run-time, on specification of a context within which to execute the interfaces, ODI will use the correct physical connections, databases + tables (source + target) linked the logical objects being used in those interfaces as defined within the environment Topology.  
  
  
**8) Does my ODI infrastructure require an Oracle database?**  
No, the ODI modular repositories (Master + and one of multiple Work repositories) can be installed on any database engine that supports ANSI ISO 89 syntax such as Oracle, Microsoft SQL Server, Sybase AS Enterprise, IBM DB2 UDB, IBM DB2/40.  
  
  
**9) Where can I get more information on ODI?**  
The OTN Data integration home page : <http://www.oracle.com/us/products/middleware/data-integration/index.html>

**10) Does ODI support web services?**  
Yes, ODI is 'SOA' enabled and its web services can be used in 3 ways:

* The Oracle Data Integrator Public Web Service, that lets you execute a scenario (a published package) from a web service call
* Data Services, which provide a web service over an ODI data store (i.e. a table, view or other data source registered in ODI)
* The ODIInvokeWebService tool that you can add to a package to request a response from a web service

**11) Where does ODI sit with my existing OWB implementation(s)?**  
As mentioned previously, the ODI-EE licence includes both ODI and OWB as separate products, both tools will converge in time into "Oracle’s Unified Data Integration Product".  
  
Oracle have released a statement of direction for both products, published January 2010:  
  
<http://www.oracle.com/technology/products/oracle-data-integrator/sod.pdf>  
  
OWB 11G R2 is the first step from Oracle to bring these two applications together, its now possible to use ODI Knowledge modules within your OWB 11G R2 environment as 'Code Templates', an Oracle white paper published February 2010 describes this in more detail:  
  
<http://www.oracle.com/technology/products/warehouse/pdf/owb-11gr2-code-template-mappings.pdf>

**12) Is ODI Used by Oracle in their products?**  
Yes there are many Oracle products that utilise ODI, but here are just a few:

* Oracle Application Integration Architecture (AIA)
* Oracle Agile products
* Oracle Hyperion Financial Management
* Oracle Hyperion Planning
* Oracle Fusion Governance, Risk & Compliance
* Oracle Business Activity Monitoring

Oracle BI Applications also uses ODI as its core ETL tool in place of Informatica , but only for one release of OBIA and when using a certain source system.  
  
Future plans are to have ODI fully available through the OBIA offering.

**Oracle Data Integrator 12c New Features:**

Here is the new features comes with Oracle Data Integrator 12 C version. Please visit this [link](http://docs.oracle.com/middleware/1212/odi/ODIDG/whatsnew.htm#ODIDG109) for more information. Let me know if have any doubts on this.

Oracle Data Integrator 12c (12.1.2) introduces the following enhancements:

1. Declarative Flow-Based User Interface

2. Reusable Mappings

3. Multiple Target Support

4. Step-by-Step Debugger

5. Runtime Performance Enhancements

6. Oracle GoldenGate Integration Improvements

7. Standalone Agent Management with WebLogic Management Framework

8. Integration with OPSS Enterprise Roles

9. XML Improvements

10. Oracle Warehouse Builder Integration

11. Unique Repository IDs

**1. what is load plans and types of load plans?**  
ANS) Load plan is a process to run or execute multiple scenarios as a Sequential or parallel or conditional based execution of your scenarios. And same we can call three types of load plans , Sequential, parallel and Condition based load plans.   
**2. what is profile in odi?**  
ANS)  profile is a set of objective wise privileges. we can assign this profiles to the users. Users will get the privileges from profile. Please refer <http://oditraining.blogspot.co.uk/2012/06/odi-security-manager-all-profiles.html>  
  
  
**3 what is the odi console?**  
ANS) ODI console is a web based navigator to access the Designer, Operator and Topology components through browser.   
  
**4.how to write the sub queries in odi?**  
ANS:)Using Yellow interface and sub queries option we can create sub queries in odi.  
or Using  VIEW we can go for sub queries Or Using ODI Procedure we can call direct DB queries  
in ODI.  
  
**5.suppose i having 6 interfaces and running the interface 3 rd one failed how to run remaining interfaces?**  
ANS: ) if you are running Sequential load it will stop the other interfaces. so goto operator and right click on filed interface and click on restart. If you are running all the interfaces are parallel only one interface will fail and other interfaces will finish.   
  
**6. how to remove the duplicate in odi?**  
ANS) Use DISTINCT in IKM level. it will remove the duplicate rows while loading into target.   
  
**7. suppose having unique and duplicate but i want to load unique record one table and duplicates one table?**  
ANS) Create two interfaces or once procedure and use two queries one for Unique values and one for duplicate values.   
  
**8. how to write the procedures in odi?**  
ANS) Procedure is a step by step any technology code operations

**1)how to implement the logic in procedures if the source side data deleted that will reflect the target side table**  
ANS)User this query on Command on target Delete from Target\_table where not exists (Select 'X' From Source\_table Where Source\_table.ID=Target\_table.ID).  
  
**2)if the src hav total 15 records with 2 records are updated and 3 records are newly inserted at the target side we have to load the newly changed and inserted records**  
ANS) Use IKM Incremental Update Knowledge Module for Both Insert n Update operations.  
  
**3)can we implement package in package?**  
ANS:) Yes we can call one package into other package.  
  
**4)how to load the data with one flat file and one rdbms table using joins?**  
ANS:) Drog n drop both File and table into source area and join as in Staging area.  
  
**5)in the package one interface got failed how to know which interface got failed if we no access to operator?**  
ANS:)Make it mail alert or check into SNP\_SESS\_LOg tables for session log details.  
  
**6)if the src and tgt are oracle technology tell me the process to achieve this requirement(interfaces,kms,models)**  
ANS) Use LKM SQL to SQL or LKM SQL to Oracle , IKM Oracle Incremental update or Control append.  
  
**7)how to implement data validations?**  
ANS:) Use Filters & Mapping Area AND DataQuality related to constraints use CKM Flowcontrol.  
  
**8)how to handle exceptions?**  
ANS) Exceptions In packages advanced tab and load plan exception tab we can handle exceptions.  
  
**9)what we specify the in xml dataserver and parameters for to connect to xml file?**  
ANS:) Filename with location :F and Schema :S this two parameters

**ODI Architecture**

1. Understand ODI Architecture.
2. Understand Components that make up ODI.
3. Understand what are ODI repositories?

Architecture Overview:

What is Oracle Data Integrator?

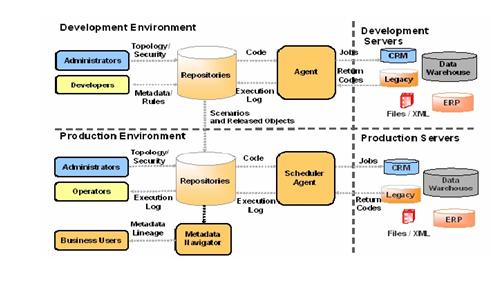
* Data integration product.
* ODI is a development platform. (Business Rule Driven , E-LT approach)
* Simple and faster.
* Based on Metadata – Centralized Repository.

Oracle Data Integrator is an integration platform. Simply put, it is used to move and transform information across the information system. Oracle Data Integrator is also a development platform for integration processes. It is unique in two respects:

* It uses an approach driven by business rules. In this approach, you focus your effort on the business side of integration, and not on the technical aspects.
* It uses the E-LT approach. Oracle Data Integrator does not execute the integration processes itself at run time, but orchestrates a process which leverages existing systems.

Oracle Data Integrator is based on metadata. That is, descriptive information about the information system and its contents. This metadata is stored in a centralized metadata repository. These elements combined mean that, Oracle Data Integrator AIP enables “Simply Faster Integration.

**ODI Architecture**

[](http://3.bp.blogspot.com/_ranZ41gSAgc/TKoSPE9SXoI/AAAAAAAAAFQ/rITOSz-6aw4/s1600/1.JPG)

The central component of the architecture is the repository. This stores configuration information about the IT infrastructure, the metadata for all applications, projects, scenarios, and execution logs. Repositories can be installed on an OLTP relational database. The repository also contains information about the Oracle Data Integrator infrastructure, defined by the administrators.

Administrators, developers, and operators use different Oracle Data Integrator Graphical User Interfaces to access the repositories.

Security and Topology are used for administering the infrastructure, Designer is used for reverse engineering metadata and developing projects, and Operator is used for scheduling and operating run-time operations.

At design time, developers work in a repository to define metadata and business rules. The resulting processing jobs are executed by the Agent, which orchestrates the execution by leveraging existing systems. It connects to available servers and requests them to execute the code. It then stores all return codes and messages into the repository.

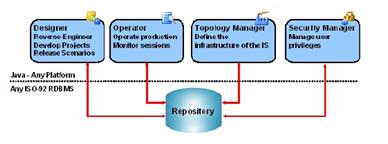
It also stores statistics such as the number of records processed, the elapsed time, and so on.

Several different repositories can coexist in a single IT infrastructure. In the graphic in the previous page, two repositories are represented: one for the development environment, and another one for the production environment. The developers release their projects in the form of scenarios that are sent to production.

In production, these scenarios are scheduled and executed on a Scheduler Agent which also stores all its information in the repository. Operators have access to this information and are able to monitor the integration processes in real time.

Business users, as well as developers, administrators and operators, can get Web-based read access to the repository. The Metadata Navigator application server links the Oracle Data Integrator Repository to any Web browser, such as Firefox or Internet Explorer

**ODI Components**

[](http://2.bp.blogspot.com/_ranZ41gSAgc/TKoS04uywwI/AAAAAAAAAFY/qCLPKT-N23g/s1600/2.JPG)

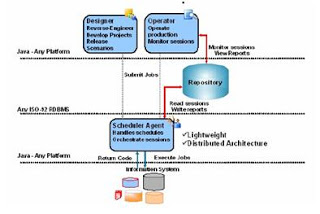
The four Oracle Data Integrator GUIs—Designer, Operator, Topology Manager, and Security Manager, are based on Java. They can be installed on any platform that supports Java Virtual Machine 1.4, including Windows, Linux, HP-UX, Solaris, pSeries, and so on.

Designer is the GUI for defining metadata, and rules for transformation and data quality. It uses these to generate scenarios for production, and is where all project development takes place. It is the core module for developers and metadata administrators. Operator is used to manage and monitor Oracle Data Integrator in production. It is designed for production operators and shows the execution logs with errors counts, the number of rows processed, execution statistics, and so on. At design time, developers use Operator for debugging purposes.

Topology Manager manages the physical and logical architecture of the infrastructure. Servers, schemas, and agents are registered here in the Oracle Data Integrator Master Repository. This module is usually used by the administrators of the infrastructure.

Security Manager manages users and their privileges in Oracle Data Integrator. It can be used to give profiles and users access rights to Oracle Data Integrator objects and features. This module is usually used by security administrators. All Oracle Data Integrator modules store their information in the centralized Oracle Data Integrator repository.

**ODI Run Time Components**

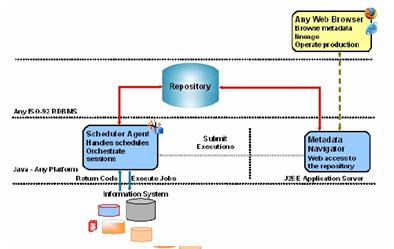
[](http://2.bp.blogspot.com/_ranZ41gSAgc/TKoT7mI3ZKI/AAAAAAAAAFo/R8FJj9fnDic/s1600/3.JPG)

At run time, the Scheduler Agent orchestrates the execution of the developed scenarios. It can be installed on any platform provided that it supports a Java Virtual Machine 1.4 (Windows, Linux, HP-UX, Solaris, pSeries, iSeries, zSeries, and so on).

Execution may be launched from one of the graphical modules, or by using the built-in scheduler. Thanks to Oracle Data Integrator’ E-LT architecture, the Scheduler Agent rarely performs any transformation itself. Normally, it simply retrieves code from the execution repository, and requests database servers, operating systems or scripting engines to execute it. When the execution is completed, the scheduler agent updates logs in the repository, reporting error messages and execution statistics.

The execution log can be viewed from the Operator graphical module. It is important to understand that although it can act as a transformation engine, the agent is rarely used this way in practice. Agents are installed at tactical locations in the information system to orchestrate the integration processes and leverage existing systems. Agents are lightweight components in this distributed integration architecture

**Metadata Navigator**

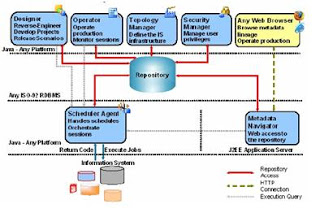
[](http://3.bp.blogspot.com/_ranZ41gSAgc/TKoUQNSiDSI/AAAAAAAAAFw/XamXmaGMm6I/s1600/4.JPG)

Metadata Navigator is a J2EE application that provides Web access to Oracle Data Integrator repositories. It allows the users to navigate projects, models, logs, and so on. By default, it is installed on Jakarta Tomcat Application Server.

Business users, developers, operators and administrators use their Web browser to access Metadata Navigator. Via its comprehensive Web interface, they can see flow maps, trace the source of all data and even drill down to the field level to understand the transformations that affected the data.

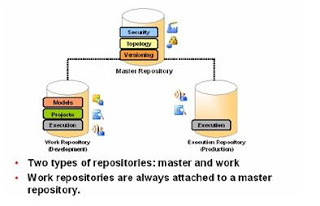
It is also possible to trigger and monitor processing jobs from a Web browser through Metadata Navigator

**Components – A global view**

[](http://3.bp.blogspot.com/_ranZ41gSAgc/TKoU4yXfndI/AAAAAAAAAF4/yVwfUMhaP0M/s1600/5.JPG)

By putting these pieces together, you now have a global view of the components that make up Oracle Data Integrator: the graphical components, the repository, the Scheduler Agent, and finally Metadata Navigator.

**ODI Repository**

[](http://1.bp.blogspot.com/_ranZ41gSAgc/TKoVKNoIxlI/AAAAAAAAAGA/1XW_UxvEAxg/s1600/6.JPG)

The Oracle Data Integrator Repository is composed of a master repository and several work repositories. These repositories are databases stored in relational database management systems. All objects configured, developed, or used by the Oracle Data Integrator modules are stored in one of these two types of repository. The repositories are accessed in client/server mode by the various components of the Oracle Data Integrator architecture.

There is usually only one master repository, which contains the following information:

· Security information including users, profiles, and access privileges for the Oracle Data Integrator platform.

· Topology information including technologies, definitions of servers and schemas, contexts and languages.

Old versions of objects. The information contained in the master repository is maintained with Topology Manager and Security Manager. All modules access the master repository, as they all need the topology and security information stored there.

The work repository is where projects are worked on. Several work repositories may coexist in the same Oracle Data Integrator installation. This is useful, for example, to maintain separate environments or to reflect a particular versioning life cycle.

A work repository stores information for:

· Data models, which include the descriptions of schemas, data store structures and metadata, fields and columns, data quality constraints, cross references, data lineage, and so on

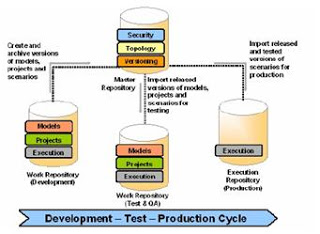
· Projects, which include business rules, packages, procedures, folders, knowledge modules, variables and so on

Execution, which means scenarios, scheduling information and logs

The contents of a work repository are managed with Designer and Operator. It is also accessed by the agent at run time.

When a work repository is only used to store execution information (typically for production purposes), it is called an execution repository. Execution repositories are accessed at run time with Operator and also by agents. An important rule to remember is that all work repositories are always attached to exactly one master repository

**Example of Repository Setup**

[](http://2.bp.blogspot.com/_ranZ41gSAgc/TKoVZw9So3I/AAAAAAAAAGI/itX-RyvkLAM/s1600/7.JPG)

This diagram gives an overview of a typical repository architecture where development, testing and production are carried out in separate work repositories. When the development team finishes working on certain projects, it releases them into the master repository. The testing team imports these released versions for testing in a separate work repository, thus allowing the development team to continue working on the next versions. When the test team successfully validates the developed items, the production team then exports executable versions (called scenarios) into the final production work repository. This repository structure corresponds to a simple development-test-production cycle.

**Declarative Flow-Based User Interface**

The new declarative flow-based user interface combines the simplicity and ease-of-use of the declarative approach with the flexibility and extensibility of configurable flows. Mappings (the successor of the Interface concept in Oracle Data Integrator 11g) connect sources to targets through a flow of components such as Join, Filter, Aggregate, Set, Split, and so on.

**Reusable Mappings**

Reusable Mappings can be used to encapsulate flow sections that can then be reused in multiple mappings. A reusable mapping can have input and output signatures to connect to an enclosing flow; it can also contain sources and targets that are encapsulated inside the reusable mapping.

**Multiple Target Support**

A mapping can now load multiple targets as part of a single flow. The order of target loading can be specified, and the Split component can be optionally used to route rows into different targets, based on one or several conditions.

**Step-by-Step Debugger**

Mappings, Packages, Procedures, and Scenarios can now be debugged in a step-by-step debugger. Users can manually traverse task execution within these objects and set breakpoints to interrupt execution at pre-defined locations. Values of variables can be introspected and changed during a debugging session, and data of underlying sources and targets can be queried, including the content of uncommitted transactions.

**Runtime Performance Enhancements**

The runtime execution has been improved to enhance performance. Various changes have been made to reduce overhead of session execution, including the introduction of blueprints, which are cached execution plans for sessions. Performance is improved by loading sources in parallel into the staging area. Parallelism of loads can be customized in the physical view of a map. Users also have the option to use unique names for temporary database objects, allowing parallel execution of the same mapping.

**Oracle GoldenGate Integration Improvements**

The integration of Oracle GoldenGate as a source for the Change Data Capture (CDC) framework has been improved in the following areas:

* Oracle GoldenGate source and target systems are now configured as data servers in Topology. Extract and replicate processes are represented by physical and logical schemas. This representation in Topology allows separate configuration of multiple contexts, following the general context philosophy.
* Most Oracle GoldenGate parameters can now be added to extract and replicate processes in the physical schema configuration. The UI provides support for selecting parameters from lists. This minimizes the need for the modification of Oracle GoldenGate parameter files after generation.
* A single mapping can now be used for journalized CDC load and bulk load of a target. This is enabled by the Oracle GoldenGate JKM using the source model as opposed to the Oracle GoldenGate replication target, as well as configuration of journalizing in mapping as part of a deployment specification. Multiple deployment specifications can be used in a single mapping for journalized load and bulk load.
* Oracle GoldenGate parameter files can now be automatically deployed and started to source and target Oracle GoldenGate instances through the JAgent technology.

**Standalone Agent Management with WebLogic Management Framework**

Oracle Data Integrator standalone agents are now managed through the WebLogic Management Framework. This has the following advantages:

* UI-driven configuration through Configuration Wizard
* Multiple configurations can be maintained in separate domains
* Node Manager can be used to control and automatically restart agents

**Integration with OPSS Enterprise Roles**

Oracle Data Integrator can now use the authorization model in Oracle Platform Security Services (OPSS) to control access to resources. Enterprise roles can be mapped into Oracle Data Integrator roles to authorize enterprise users across different tools.

**XML Improvements**

The following XML Schema constructs are now supported:

* list and union - List or union-based elements are mapped into VARCHAR columns.
* substitutionGroup - Elements based on substitution groups create a table each for all types of the substitution group.
* Mixed content - Elements with mixed content map into a VARCHAR column that contains text and markup content of the element.
* Annotation - Content of XML schema annotations are stored in the table metadata.

**Oracle Warehouse Builder Integration**

Oracle Warehouse Builder (OWB) jobs can now be executed in Oracle Data Integrator through the OdiStartOwbJob tool. The OWB repository is configured as a data server in Topology. All the details of the OWB job execution are displayed as a session in the Operator tree.

**Unique Repository IDs**

Master and Work Repositories now use unique IDs following the GUID convention. This avoids collisions during import of artifacts and allows for easier management and consolidation of multiple repositories in an organization.

**Using the Oracle Data Integrator Tools**  
  
  
Oracle Data Integrator Tools (**also called Oracle Data Integrator Commands)** are commands provided for performing specific tasks at run-time. These tasks may be as simple as waiting for a certain time or producing a sound, or as sophisticated as executing ANT Scripts or reading emails from a server.   
  
They are used in Packages, in Procedure Commands, in Knowledge Modules Commands or directly from a command line.   
  
Note: Previous versions of Oracle Data Integrator that supported calling built-in tools from Jython or Java scripts using their internal Java classes (such as SnpsSendMail and SendMail). The usage of tools using this method is deprecated since version 10.1.3.2.0 and should be avoided.   
  
**Note: The carriage return in a command is not allowed.**   
**Using a Tool in a Package**  
  
How to add and use an Oracle Data Integrator Tool in a Package is covered in Adding Oracle Data Integrator Tool Steps.   
  
It is possible to sequence the tools steps within the package, and organize them according to their success and failure. For more information, refer to Defining the Sequence of Steps and Arranging the Steps Layout.   
  
In a package, it possible to use directly in the tool parameters variable values, sequences or Oracle Data Integrator substitution method calls. Refer to Working with Procedures, Variables, Sequences, and User Functions for more information.   
  
**Using a Tool in a Knowledge Module or a Procedure Command**  
  
How to use an Oracle Data Integrator Tool in a KM or Procedure is covered in Working with Procedures.   
  
In a knowledge module or a procedure, it possible to use directly in the tool parameters variable values, sequences, Oracle Data Integrator substitution method calls or the results from a SELECT statement. Refer to Working with Procedures, Variables, Sequences, and User Functions for more information.   
  
**Using a Tool from a Command Line**  
  
Note: The command line scripts, which are required for performing the tasks described in this section, are only available if you have installed the Oracle Data Integrator Standalone Agent. See the Oracle Fusion Middleware Installation Guide for Oracle Data Integrator for information about how to install the Standalone Agent.   
  
**To use an Oracle Data Integrator Tool from a command line:**   
Launch a Shell (UNIX), a Command Prompt (Windows).   
  
Go to the oracledi/agent/bin sub-directory of the Oracle Data Integrator installation directory.   
  
Launch the startcmd.bat (Windows) or startcmd.sh (UNIX) command, with the following syntax:   
startcmd <command\_name> [<command\_parameters>]\*  
  
Note: On Windows platforms, it is necessary to surround the command arguments containing "=" signs or spaces, by using double quotes. The command call may differ from the UNIX command call. For example:   
startcmd.bat OdiSleep "-DELAY=5000" (Windows)  
./call startcmd.sh OdiSleep -DELAY=5000 (UNIX)

The <command\_name> parameter is case-sensitive.

Oracle Data Integrator Interview Questions and answers?  
  
1) What is the history of ODI?  
2) What are the types of Knowledge Modules?  
3) What are the types of data contol?  
4) What is Journalization and why we are using in ODI?  
5) Tell me step by step to enable Journalization?  
6) What is An Interface ?  
7) What is temporary Interface( Yellow Interface) ?  
8) Tell me some differences between ODI 10g and ODI 11g?  
9) Tell what are the ODI tools have u used in your project?  
10) What are the Types of repositories in ODI?  
11) Can i create more than one Master Repository in ODI?  
12) What is a scenario in odi?  
13) What is load plan and types of load plans in ODI?  
14) What is CKM and when we will use this CKM?  
15) What is SKM and when we will use this SKM?  
16) What is package and main advantages?  
17) Differences between package and loadplans?  
18) How to load data from multiple files to single target?  
19) How to load data from file to file and what are the KM's required for this requirement?  
20) how to pass a variable more than one values?   
21) What is an agent and What are the types of agents?  
22) What is StandAlone Agent and default port number?  
23) What is J2EE Agent and port number?  
24) What is the use solutions?  
25) What is flexfileds and use of flexfileds in odi?  
26) What is CDC and explain complete CDC process flow?  
27) What are the types of profiles in ODI?  
28) What are the types of Generic profiles types in ODI?  
29) Which profile required to access Models in Designer tool?  
30) Which profile required to access solution in Designer tool?  
31) Which profile required to access ODI Console?  
32) What is ODI console?  
33) What are types of LOG LEVELS in ODI?  
34) What is the use of Markers?  
35) What is memo?  
36) How to move objects from DEV to QA and QA to PROD Environments?