

**Mumbai University**

**NOTES**

**B.Sc.IT: SEMESTER – VI**

**AN INTRODUCTION  
TO ORACLE  
WAREHOUSE  
BUILDER**

**[CBSGS – 75:25 Pattern]**

**DATA**

**WAREHOUSE**

# 1

## An Introduction to Oracle Warehouse Builder

### Introduction to data warehousing

Although you may not be familiar with data warehousing, you have probably at least heard the term. Data warehouses are becoming increasingly common as businesses have realized the need to be able to mine the information they have stored in the electronic form in order to provide a valuable insight into the operation of their business and how best to improve it. **Organizations need to monitor these processes, define policies, and—at a more strategic level—define the visions and goals that will move the company forward in the future. Operational transactional systems have greatly benefited the daily functioning of the enterprise. But now, organizations are shifting to a more decisional-based requirement from their computing platforms and are looking to build data warehouses. This is where OWB enters the picture to help organizations with the task of building that data warehouse.**

## Introduction to our fictional organization

The manuals that Oracle supplies with its database and applications contain a great deal of information. However, it can be hard to relate that information to the real-world ways of implementing the database and applications. Anyone who has ever tried to read a technical user guide or reference provided with a database or application will know what that means. It is a great benefit to be able to learn about a new software tool by seeing how that tool is actually used within the context of an actual organization conducting a business. This is precisely the focus of this book. We'll be building an actual data warehouse using a fictional organization as an example.

Before we talk about what a data warehouse is, let's get introduced to the fictional organization we'll be using to demonstrate the use of the Warehouse Builder to build a data warehouse. **Throughout this book, we will be using examples of the concepts involved by making reference to a fictional organization named ACME Toys and Gizmos, which is sales oriented. It is an entirely made-up organization, and any similarity to a real company is completely coincidental. This book will provide explanations throughout on how to use the OWB tool to build a data warehouse within the context of this invented company, which is involved in storefront and online Internet sales. Thus, it will demonstrate practical ways of implementing a data warehouse that can be directly applied in the real world.**

**ACME Toys and Gizmos will have stores all over the United States as well as a number of other countries, and will also have an online storefront for Internet sales. The online transactional processing systems (OLTP) play a huge role in the functioning of any business today, especially in the operation of a sales-oriented business. So this makes a good example to illustrate the subject matter of data warehousing and how to take information from those OLTP systems to load our warehouse.**

Although we'll be using a sales organization for our examples, the concepts we'll discuss can apply to any business and will be as generic as possible to assist in doing that.

## What is a data warehouse?

We've discussed the business case for implementing a data warehouse by showing how companies these days need information to support strategic-level decision making. We've also introduced the fictional organization that we'll use to provide examples of the concepts we'll be presenting. But we've not yet explained what a data warehouse is.

---

We will not be dealing in detail with the concept of a data warehouse as that topic would encompass the entire contents of a book by itself. There are a number of good books already written about that topic. Therefore, we will touch upon some high-level concepts only as an introduction and to provide a context for using ODB to build a data warehouse.

**Fundamentally, a data warehouse is a decisional database system. It is designed to support the decision makers in the organization in ways a transactional processing system is ill-equipped to handle, such as the strategic-level goals and visions of an organization. To think strategically, a large amount of data over long periods of time is needed. Transactional systems are concerned with the day-to-day operations such as: How many dolls did we sell today and will we need to restock the inventory? How many orders were processed today? How many balls were shipped out today? The strategic thinkers are more concerned with questions such as: How many dolls did we sell today compared to the same time period in the last year? How has our inventory level been for the last few months?**

**To support that level of information, we need more data than what is provided by the day-to-day transactions. We'll need much more information compiled over greater time periods and this is where the data warehouse comes in. As a data warehouse is different from a transactional database, there are some unique terms used to describe the data it contains. There are also other techniques that should be employed for designing the database for a data warehouse, which would not be a good idea for a transactional database.**

**The data in a data warehouse is composed of facts (actual numerical measures) and dimensions (descriptive data about those measures) that place the facts in a context that is understandable to the end-user decision maker. For instance, a customer makes a purchase of a toy with ACME Toys and Gizmos on a particular day over the Internet, which results in a dollar amount of the transaction. The dollar amount becomes the fact and the toy purchased, the customer, and the location of the purchase (the Internet in this case) become the dimensions that provide a scope of the fact measurement and give it a meaning.**

**The design of a data warehouse should be different from that of a transactional database. The data warehouse must handle large amounts of data, and must be simple to query and understand by the end users. While relational techniques and normalization are excellent database design methods for transactional systems to ensure data integrity, they can make understanding a data warehouse difficult for the end users. They can also bog down a data warehouse with long-running queries that have to make use of many joins (including more than one table that share a common data element to look up additional data).**

---

A much better means of representing the data is to de-normalize the data, so that users will not have to be concerned with retrieving the data from multiple tables.

The use of foreign keys (a column that references a row in another table) should be restricted in a data warehouse. The outcome is a fact table with foreign keys only to each of the dimension tables. The diagram of the database structure has a fact table in the middle surrounded by dimension tables, resulting in something that looks like a star. Thus, the term star schema is used to refer to this representation of a data warehouse. It is also possible that these dimensions may themselves have other tables surrounding them, resulting in something akin to a snowflake. Thus, the term snowflake schema is also used. This is the dimensional modeling technique of representing a data warehouse.

This design lends itself extremely well to the task of querying large amounts of data by the end users. Users do not have to be bothered with queries involving complicated joins with multiple tables to get the descriptive information they need. This is because the information is included directly in the dimension tables in a de-normalized fashion. If a manager for ACME Toys and Gizmos needs to know what products sold well in the last quarter, the query will only involve two tables—the main fact table containing the data on number of items sold and the product dimension table that contains all the information about the product. The de-normalization means the manager will not have to be concerned with looking up product information in any other tables, as all the details about the product will be included in the one dimension table.

All this is great background information on data warehouses, but you can read any number of other books for much more detailed material on the topic. Our purpose in this book is to introduce the Oracle Warehouse Builder and use it to design and build our first data warehouse. So, let's see how it fits in to this discussion of data warehousing.

## **Where does OWB fit in?**

The Oracle Warehouse Builder is a tool provided by Oracle, which can be used at every stage of the implementation of a data warehouse, from initial design and creation of the table structure to the ETL process and data-quality auditing. So, the answer to the question of where it fits in is—everywhere. It is provided as a part of the Oracle Database Release 11g installation. For the previous Oracle Database Releases, it can be downloaded and installed from Oracle's web site as a free download.

---

We can choose to use any or all of the features as needed for our project, so we do not need to use every feature. Simple data warehouse implementations will use a subset of the features and as the data warehouse grows in complexity, the tool provides more features that can be implemented. It is flexible enough to provide us a number of options for implementing our data warehouse as we'll see in the remainder of the book.

## Installation of the database and OWB

We'll be using the latest version of the database as of this writing—*Oracle Database 11g Release 1*—and the corresponding version of OWB that (as of this release) is included with the database install. If you have that version of the database installed already, you can skip this section and move right on to the next. If not, then keep reading as we discuss the installation of the database software.

### Downloading the Oracle software

We can download the Oracle database software from Oracle's web site, provided we adhere to their license agreement. This agreement basically says we agree to use the database and the accompanying software either for development of a prototype of our application or for our own learning purposes. If we proceed to use this application internally or make it commercially available, then we will need to purchase a license from Oracle. For the purpose of working through the contents of this book to learn OWB, we need to install the database, which is covered under the license agreement for the free download.

**We can find the database on the Oracle Technology Network web site (<http://www.oracle.com/technology>). The main database download is usually the first download listed under FEATURED DOWNLOADS on the main page. We need to register on the site, in order to create an account, before it lets us download any files, but there is no charge for that. The download files are classified by the platform on which they can be executed, so we'll choose the one for the system we'll be hosting the database on. We'll have to accept the license agreement first before the web page will let us download the file. The download files are anywhere from 1.7 GB to 2.3 GB in size, depending on the platform we'll be hosting it on. So we do not want to attempt this download unless we have a Broadband Internet connection (that is, cable, DSL, and so on). We'll download the install file and unzip it to a folder on a drive with enough available space. The installation files are temporary and are not needed after the installation is done, so we'll be able to delete them to free up space if needed.**

---

## A word about hardware and operating systems

When installing software of this magnitude, we have to decide whether we'll have to buy additional hardware and a different operating system to run the database and OWB. OWB will run in the following databases:

- Oracle Database 11g R1 Standard Edition
- Oracle Database 11g R1 Enterprise Edition
- Oracle Database 10g R2 Standard Edition
- Oracle Database 10g R2 Enterprise Edition

This list is for the most recent version of OWB, which we'll be using throughout this book. We can download older versions of OWB that will run on older versions of the database, but we will not have the benefit of the improvements as in the latest version of the software. Much of what we'll be doing with the software throughout the course of the book can also be done on previous versions of the software.

However, due to the changes made to things such as the interface, it would be easiest to follow along using the most recent version.

For this book, the platform is Windows Vista with Oracle Database 11g Release 1 (11.1.0.6) Enterprise Edition (which is the most recent version as of this writing), which is available from the download site. The Enterprise Edition of the database was chosen because it allows us to make full use of the features of the Warehouse Builder, especially in the area of dimensional modelling. There are some errors that will be generated by the client software when running in the Standard Edition installation due to code dependencies. These code dependencies are in libraries that are installed with the Enterprise Edition, but not the Standard Edition. We could use OWB with the Standard Edition, but then we would be limited in the type of objects we could deploy. For instance, dimensions and cubes would be problematic, and without using them we'd be missing out on a major functionality provided by the tool. If we want to develop any reasonably-sized data warehouse, the Enterprise Edition is the way to go.

**Everything that we'll work through in this book was done on a laptop personal computer with an Intel Core 2 processor running at 1.67 GHz and 2 GB of RAM. Oracle says 1 GB of RAM will suffice, but it is always good to have more to provide better performance. Minimum specifications usually result in underpowered systems for all but the very basic processing. In terms of hard disk space, Oracle specifies that 4.5 GB is required for the basic database installation. We'll need about 2 GB just to save the installation files, so to make sure we have plenty of space, we should plan for something between 10GB and 15GB of available disk space just to be safe. We don't want to install the database software and then find that we don't have any space on our hard drive.**

---

**Oracle supports its database installed on Windows and Unix.** For Windows, it supports Windows XP Professional or Windows Vista (Business Edition, Enterprise Edition, or Ultimate Edition) as well as Windows Server 2003. The system mentioned above that was used for writing this book and working through all the examples, is running Windows Vista Home Premium Edition with Service Pack 1 and the database installed runs on it. We certainly would not want to use this configuration for large production databases, but it works fine for simple databases and learning purposes. The installation program will first do a prerequisite check of the computer and will flag any problems that it sees, such as not enough memory or an incorrect version of the operating system. For working through this book on our own to learn about the Warehouse Builder, we should be OK as long as we are running XP or Vista. However, for business users who would be installing the Oracle Database and OWB for use at work using Windows, it would be a good idea to stick with the recommended configurations of Windows XP Professional, Windows Vista (Business Edition, Enterprise Edition, and Ultimate Edition), or Windows Server.

—


## Installing Oracle database software

**So far we've decided what system we're going to host the database on, downloaded the appropriate install file for that system, and unzipped the install files into a folder to begin the installation. We'll navigate to that folder and run the `setup.exe` file located there. This will launch the Oracle Universal Installer program to begin the installation.**

---



**We are installing the full database, which now automatically includes the Warehouse Builder client and database components. If we had an older version of the database (10g R2 for example) that did not include the Warehouse Builder software, or if we wanted to run the client on a different workstation than where the database software is installed, then there is the option to install the Warehouse Builder by itself.**

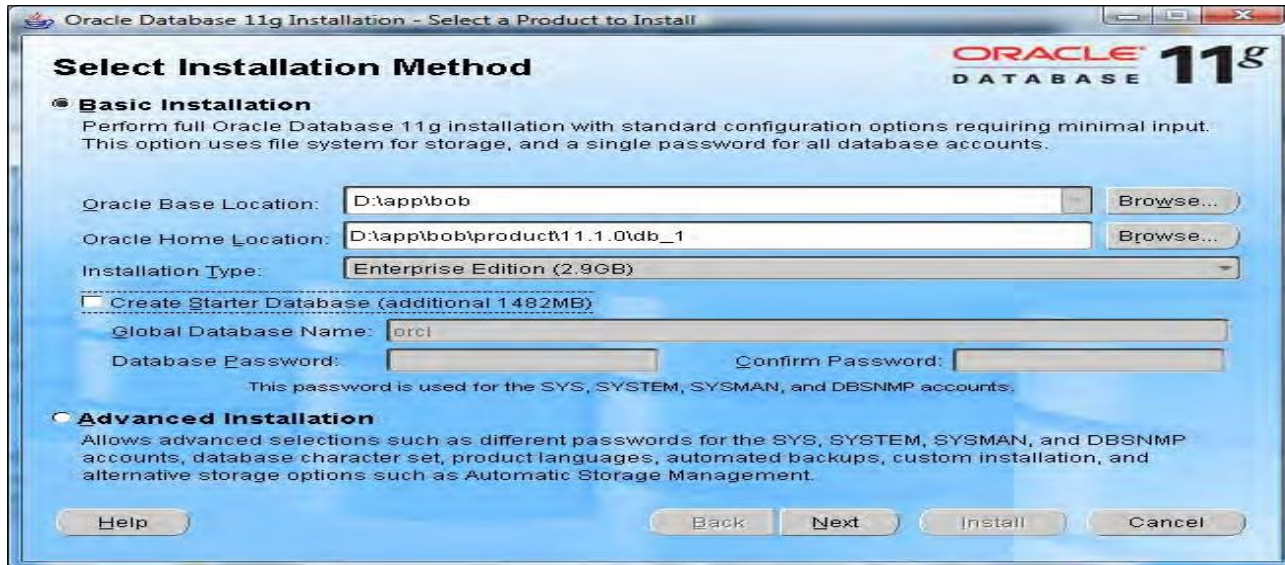
[  A separately downloadable install for the standalone option is available at <http://www.oracle.com/technology/software/products/warehouse/index.html>. Skip ahead to the section titled *Installing the OWB standalone software* if just the Warehouse Builder software is needed. ]

**One of the first questions the installer will ask us is about setting up our ORACLE\_HOME—the destination to install the software on the system and the name of the home location. Oracle uses this information when running to determine where to find its files on the system. It will store the information in the registry on Windows. It will suggest a default name, which can be changed. We'll leave it set to the default—OracleDb11g\_home1.**

**The ORACLE\_BASE and ORACLE\_HOME locations will have suggested paths filled in for us. It is a good idea to leave the path names as they are and only change the drive designation if we'd like to install to a different hard drive. The install program will suggest a drive for the installation, but we might have a different preference.**

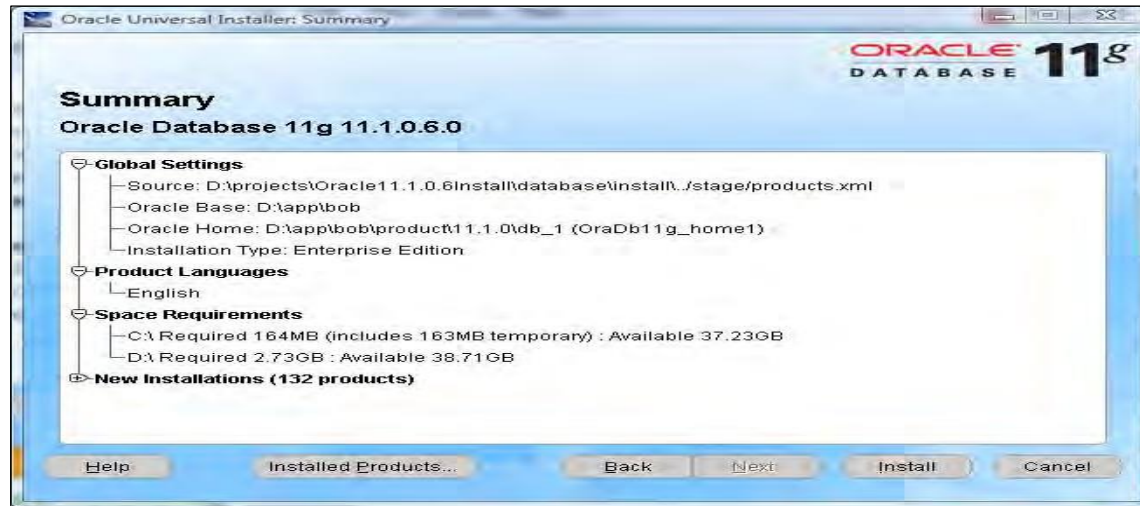
Oracle recommends a convention for naming folders and files that they call the **Optimal Flexible Architecture (OFA)**. This is described in *Appendix B* of the *Oracle Database Installation Guide* for Microsoft Windows, which can be found at the following URL: [http://download.oracle.com/docs/cd/B28359\\_01/install.111/b32006/ofa.htm#CBBEDHEB](http://download.oracle.com/docs/cd/B28359_01/install.111/b32006/ofa.htm#CBBEDHEB). It is a good idea to follow their recommendations for standardization so that others who have to work with the database files will know where to find them, and to save us from problems with possible conflicts with other Oracle products we may have installed. If we keep the default folder locations intact and only change the drive letter, we will adhere to the standard. We'll be asked to choose our installation method and whether to install a starter database. We're not going to let it install a starter database for us because it's going to default to a transactional database and we want a data warehouse. **So on the Select Installation Method screen, we'll check off the Basic Installation type and uncheck the box for installing a starter database. The Select Installation Method screen should look similar to the following:**

---

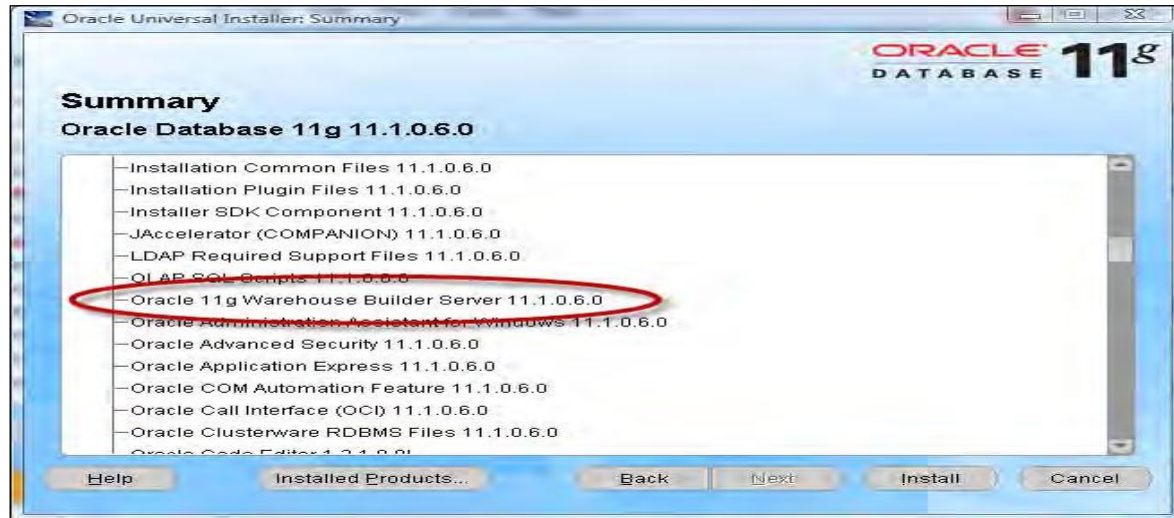


We will click on the **Next** button to continue and the install program will perform its prerequisite checks to ensure our system is capable of running the database. That should show a status of **Succeeded** for all the checks. If any of the checks do not pass, we have to correct them and start over before continuing. When everything reports a status of **Succeeded**, we can click on the **Next** button.

The install screens will proceed to the **Summary** screen where we can verify the options that we selected for the installation before actually doing it. So here we can make any last minute changes.



If we expand the **New Installations** entry, it will list all of the database products and features that will be installed. This includes the feature we are most interested in, the **Oracle 11g Warehouse Builder Server** option, which is included automatically in 11g database installations. The following image illustrates what will appear in the list for OWB and the option we are interested in is circled:



When it completes we'll be presented with the final screen with a little reminder similar to the following where bob is the login name of the user running the installation:



This is important information because our database could be rendered inoperable if files are deleted. Now that it's installed, it's time to proceed with creating a database. But there is one step we have to do first—we need to configure the **listener**.

## Configuring the listener

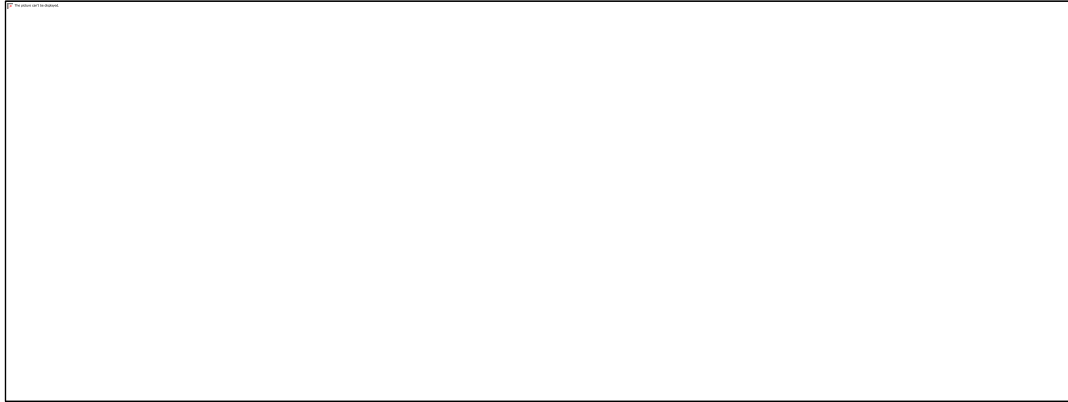
The listener is the utility that runs constantly in the background on the database server, listening for client connection requests to the database and handling them. It can be installed either before or after the creation of a database, but there is one option during the database creation that requires the listener to be configured—so we'll configure it now, before we create the database.

**Run Net Configuration Assistant to configure the listener. It is available under the Oracle menu on the Windows Start menu as shown in the following image**



The welcome screen will offer us four tasks that we can perform with this assistant. We'll select the first one to configure the listener, as shown here:

---



The next screen will ask you what we want to do with the listener. The four options are as follows:

- **Add**
- **Reconfigure**
- **Delete**
- **Rename**

Only the **Add** option will be available since we are installing Oracle for the first time. The remainder of the options will be grayed out and will be unavailable for selection. If they are not, then there is a listener already configured and we can proceed to the next section—*Creating the database*.

For those of us installing for the first time on our machines, we need to proceed with the configuration. The next screen will ask us what we want to name the listener. It will have **LISTENER** entered by default and that's a fine name, which states exactly what it is, so let's leave it at that and proceed.

The next screen is the protocol selection screen. It will have **TCP** already selected for us, which is what most installations will require. This is the standard communications protocol in use on the Internet and in most local networks. Leave that selected and proceed to the next screen to select the port number to use. The default port number is 1521, which is standard for communicating with Oracle databases and is the one most familiar to anyone who has ever worked with an Oracle database. So, change it only if you want to annoy the Oracle people in your organization who have all memorized the default Oracle port of 1521.

### To change or not change the default listener port



Putting aside the annoyance, the Oracle people might have to suffer as there are valid security reasons why we might want to change that port number. Since it is so common, the people accustomed to working with the Oracle database aren't the only people who know that port number. Hackers looking to break into an Oracle database are going to go straight for that port number, so if we change it to something obscure, the database will be harder to find on the network for the people with malicious intent. If it does get changed, be sure to make a note of the assigned number.

There also may be firewall issues that allow only certain port numbers to be open through the firewall, which means communication on any of the other port numbers would be blocked. 1521 might be allowed by default since it is common for the Oracle database. It would be a good idea to check with the network support personnel to get their recommendation.

That is the last step. It will ask us if we want to configure another listener. Since we only need one, we'll answer "no" and finish out the screens by clicking on the **Finish** button back on the main screen.

## Creating the database

So far we have the Oracle software installed and a listener configured, but we have not created a database. We chose not to install the starter database because that defaults to a general purpose transactional database, and we want one that is oriented toward a data warehouse.

We can install a new database using **Database Configuration Assistant**, which Oracle provides to walk us step-by-step through the process of creating a database. It is launched from the Windows **Start** menu as shown in the following image:



Running this application may require patience as we have to wait for the application to load after it's selected. Depending on the system it is running on, it can take over a minute to display, during which there is no indication that anything is happening. It may be tempting to just select it again from the **Start** menu because it appears it didn't work the first time, but don't as that will just end up running two instances of the program. It will appear soon. The following are steps in the creation process:

1. **The first step is to specify what action to take. Since we do not have a database created, we'll select the Create a Database option in Step 1. If there was a database already created, the options for configuring a database or deleting a database would be selectable.** Templates can be managed with the **Database Configuration Assistant** application, which are files containing preset options for various database configurations. Pre-supplied templates are provided with the application, and the application has the ability to custom-build templates.

Automatic Storage Management can be configured as well. It is Oracle's feature for databases for automatically managing the layout and storage of database files on the system. These are both topics for a more advance book on the Oracle Database. We will be creating a database using an existing template.

2. This step will offer the following three options for a database template to select:
  - **General Purpose or Transaction Processing**
  - **Custom Database**
  - **Data Warehouse**

We are going to choose the **Data Warehouse** option for our purposes. If we already had a database installed that we wanted to use for learning OWB, but that's not configured as a data warehouse, it's not a problem. We can still run OWB hosted on it and create the data warehouse schema (database user and tables), which we'll be creating as we proceed through the book. This would be fine for learning purposes, but for production-ready data warehouses a database configured specifically as a data warehouse should be used.

3. **This step of the database creation will ask for a database name. The name of the database must be one to eight characters in length.** Any more than that will generate an error when trying to proceed to the next screen. This is an Oracle database limitation. The database name can also include the network domain name of the domain of the host it is running on, to further uniquely identify it. Follow the name with a period and then the domain, which itself can include additional periods.
-

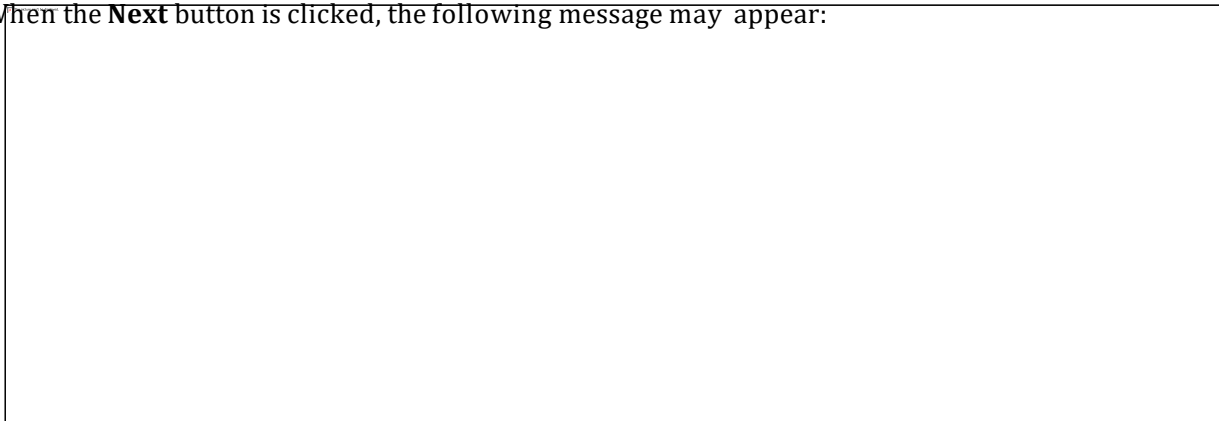


If this database is being created for business use, a good naming scheme would reflect the purpose of the database. Since we're creating this database for the data warehouse of ACME Toys and Gizmos Company, we'll choose a name that reflects this—ACME for the company name and DW for data warehouse, resulting in a database name of ACMEDW. It is important to remember this name as it will be a part of any future connections to the database.

As the database name is typed in, the **SID** (or Oracle **System Identifier**) is automatically filled in to match it. If the domain is added to the database name, the SID will stop pre-populating after the first period is entered. The end result is that the SID becomes the same as the first part of the database name.

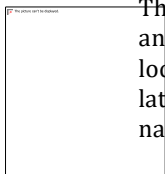
4. This step of the database creation process asks whether we want to configure **Enterprise Manager**. The box is checked by default and leave it as is. This is a web-based utility Oracle provides for controlling a database, and as it is very useful to have, we will want to enable it. There are two options for controlling a database: registering with **Grid Control** or local management. Grid Control is Oracle's centralized feature for controlling a grid, a network of loosely coupled modular hardware and software components that can be joined and rejoined together on demand to meet business needs. That is what the "g" in Oracle Database 11g stands for. If your network is not configured in a grid architecture, or you are installing on a standalone machine, then choose the local management option. It will automatically detect a Grid Control agent that is running locally, and if it doesn't find one, the **Grid Control** option will be grayed out anyway. In that case, you will only be able to select local management.

When the **Next** button is clicked, the following message may appear:



That means a listener was not configured before creating the database. If this happens, we'll have to just pause our database creation and go back to the previous section about installing the listener and then come right back to this spot. There is no need to exit out of the database install window while doing this; just leave it on step 4. When we've completed the listener configuration, this screen will allow us to proceed to the next screen without that warning popping up again.

5. On this screen (step 5) we can set the database passwords on the system accounts using a different one for each account, or by choosing one password for all four. We're going to set a single password on all four, but for added security in a production environment, it is a good idea to make a different password for each. Click on the option to **Use the Same Administrative Password for All Accounts** and enter a password. This is very important to remember as these are key system accounts used for database administrative control.
6. This step is about storage. We'll leave it at the default of **File System** for storage management. The other two options are for more advanced installations that have greater storage needs.
7. This step is for specifying the locations where database files are to be created. This can be left at the default for **simplicity** (which uses the locations specified in the template and follows the OFA standard for naming folders described above). A storage screen will come up where we'll be able to change the actual file locations if we want, for all but the **Oracle-Managed Files** option.



The Oracle-Managed Files option is provided by the database so that we can let Oracle automatically name and locate our data files. A folder location is specified on the step 7 screen, which will become the default location for any files created using this option. This is why we won't be able to change any file locations later on during the installation if this option is chosen. However, files can still be created with explicit names and locations after the database is running.



8. The next screen is for configuring recovery options. We're up to step 8 now. If we were installing a production database, we would want to make sure to use the **Flash Recovery** option and to **Enable Archiving**. Flash Recovery is a feature Oracle has implemented in its database to provide a location that is managed by the database. It stores backups and files needed to recover a database in the event of disk failure. With **Flash Recovery Area** specified, we can recover data that would otherwise be lost in a system failure.

Enabling archiving turns on the archive log mode of the database, which causes it to archive the **redo logs** (files containing information that is used by the database to recover transactions in the event of a failure.) Having redo logs archived means you can recover your database up to the time of the failure, and not just up to the time of the last backup.

These recovery options will consume more disk space, but will provide a recovery option in the event of a failure. Each individual will have to make the call for their particular situation whether that is needed or not.

We'll specify **Flash Recovery** and for simplicity, we will just leave the default for size and location. We will not enable archiving at this point. These options can always be modified after the database is running, so this is not the last chance to set them.

9. This step is where we can have the installation program create some sample schemas in the database for our reference, and specify any custom scripts to run. The text on the screen can be read to decide whether they are needed or not. We don't need either of these for this book, so it doesn't matter which option we choose.
  10. The next screen is for **Initialization Parameters**. These are the settings that are put in place to define various options for the database such as **Memory** options. There are over 200 different parameters and to go through all of them would take much more time and space than we have here. There is no need for that at this point as there are about 28 parameters that Oracle says are basic parameters that every database installation should set. We're just going to leave the defaults set on this screen, which will set the basic parameters for us based on the amount of memory and disk space detected on our machine. We'll just move on from here. Once again, these can all be adjusted later after the database is created and running if we need to make changes.
  11. The next screen is for **security settings**. For the purposes of this book and its examples, we'll check the box to **Revert to pre-11g security settings** since we don't need the additional features. However, for a production environment, it is a good idea to leave the default checked to use Oracle's more advanced security features.
  12. This step is **automatic maintenance and we'll deselect that option and move on**, since we don't need that additional functionality. **Automatic Maintenance Tasks** are tasks that run in predefined maintenance windows of time to perform various preconfigured maintenance operations on the database. Since the database for this book is only for learning purposes, it is not critical that these maintenance tasks be done automatically.
-

Automatic maintenance is designed to run during preset maintenance windows, which are usually in the middle of the night. So if the database system is shut down every day, there wouldn't be a good window to run the tasks on regularly anyway. If installing in a production environment with servers that will be running 24 hours a day every day, then consider setting up the automatic maintenance to occur. Oracle provides three pre-configured maintenance tasks to choose from—collecting statistics for the query optimizer (for improving performance of SQL queries), **Automatic Segment Advisor** for analyzing storage space for areas that can possibly be reclaimed for use, and the **Automatic SQL Tuning Advisor** for automatically analyzing SQL statements for performance improvements.

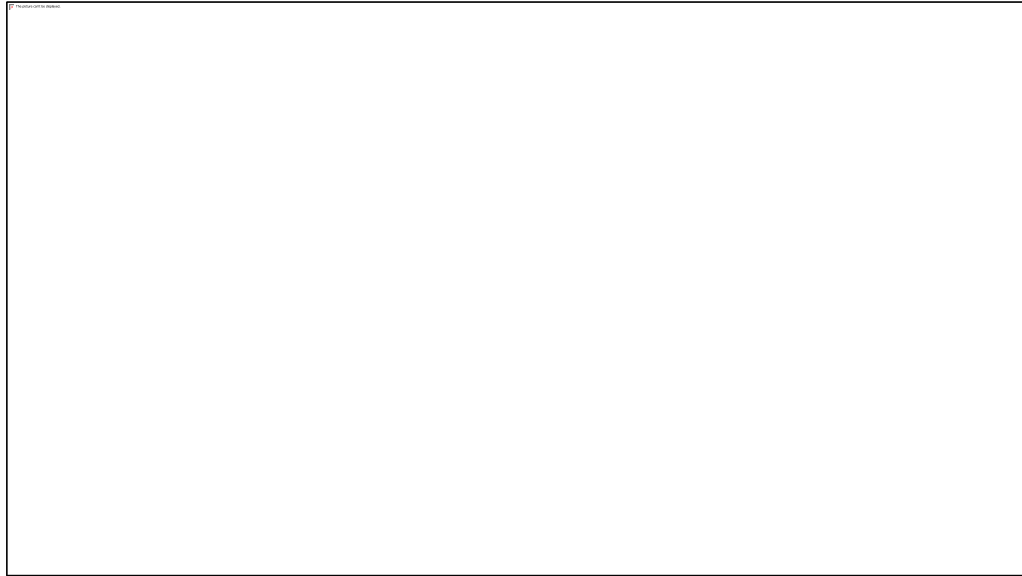
13. **The next step (step 13 of 14) is the Database Storage screen referred to earlier. Here the locations of the pre-built data and control files can be changed if needed. They should be left set to the default for simplicity since this won't be a production database.** For a production environment, we would want to consider storing datafiles on separate partitions for performance reasons, and to minimize the impact of disk failures on the running database if something goes wrong. If all the datafiles are on one drive and it goes bad, then the whole database is down.
14. **The final step has the following three options, and any or all can be selected for creating the database:**
  - **Create the database directly**
  - **Save the creation options as a template for later use**
  - **Save database creation scripts that can be used later to create the database**

**We'll leave the first checkbox checked to go ahead and create the database.**

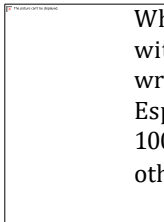
**The Next button is grayed out since this is the last screen. So click on the Finish button to begin creating the database using the selections we've just chosen. It will display a summary screen showing what options it will be using to install with. We can save this as an HTML file if we'd like to keep a record of it for future reference.**

---

All that information will be available in the database by querying system tables later, but it's nice to have it all summarized in one file. We can scroll down that window and verify the various options that will be installed, including **Oracle Warehouse Builder**, which will have a **true** in the **Selected** column as shown here:



We will be presented with the progress screen next that will show us the progress as it creates the database.



When the install progress screen gets to 100% and all the items are checked off, we will be presented with a screen summarizing the database configuration details. Take a screen capture of this screen or write down the details because it's good to know information on how the database is configured. Especially, we'll need the database name in later installation steps. We may see the progress screen at 100% doing nothing with apparently no other display visible. Just look around the desktop underneath other windows for the **Database Configuration Screen**. It's important for the next step.



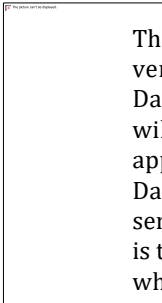
On the final Database Configuration Screen, there is a button in the lower right corner labeled **Password Management**. We need to click on this button to unlock the schema created for OWB use. Oracle configures its databases with most of the pre-installed schemas locked, and so users cannot access them. It is necessary to unlock them specifically, and assign our own passwords to them if we need to use them. One of them is the **OWBSYS** schema. This is the schema that the installation program automatically installs to support the Warehouse Builder. We will be making use of it later when we start running OWB. Click on the **Password Management** button and on the resulting Password Management screen, we'll scroll down until we see the **OWBSYS schema** and click on the check box to uncheck it (indicating we want it unlocked) and then type in a password and confirm it as shown in the following image:



Click on the **OK** button to apply these changes and close the window. On the Database Configuration Screen, click on the **Exit** button to exit out of the Database Configuration Assistant.

That's it. We're done installing our first database and it's ready to use. Next, we'll discuss installing the OWB client if we want to run the client on another computer, or if we already have a 10gR2 database installed that we want to use with the Warehouse Builder.

---



There is a known bug that can occur when running Oracle Database Control version 11.1.0.6. This is the version that we're installing for this book. It occurs only on Windows Vista 32-bit and affects only the Database Control application, not the database itself. While the Database Control service is running, you will occasionally see a popup with the message **nmefwmi.exe has stopped working**. This is apparently a rather harmless error. To keep it from occurring, you can stop the Enterprise Manager Database Control service. It is accessible from the **Administrative Tools | Services** menu entry, and is the service named **OracleDBConsoleACMEDW**, where the **ACMEDW** is the SID for the database. If you need to use the Database Control application, you can start it up, do what you need to do, and then stop it. Oracle has reported that this bug is fixed in version 11.1.0.7.

## Installing the OWB standalone software

If we are going to run the OWB client on the same computer as we just installed the Oracle database on, we don't need any more installations. That is the configuration used in this book. The OWB client software is now installed by default with the main database installation. We can verify that by checking the **Start** menu entry for Oracle. We will see a submenu entry for **Warehouse Builder** as shown in the following image:



If we want to run the OWB client on another computer on the network, or if we have an older version of the database already installed (10g Rel 2) and want to be able to use the Warehouse Builder software with it, we'll need to continue here with the installation of the OWB client software. For all others, we can proceed to the next section on OWB—*OWB components and architecture*.

For the task of installing the standalone client, we'll need to download the OWB client install file. So we will go back to the Oracle site on the Internet. The download page is at the following URL at the time of writing: <http://www.oracle.com/technology/software/products/warehouse/index.html>. If that is not working, go to the main Oracle site and search for the **Business Intelligence | Data Warehousing** page where there is a link for the download of the OWB client.

Once again we'll have to accept the license agreement before the download links will become active. So we'll accept it and download the install file to the client computer on which we'll be installing the software. The Windows ZIP file is about 1.1 GB in size so we need to make sure we have enough room on our hard drive to store the file. We'll need at least double that amount of space because the install files will take up that much space when unzipped.

When we have downloaded the ZIP file and unzipped it to our hard drive, run `setup.exe` in the top-level folder to run the Oracle Universal Installer. It should look familiar. Oracle is definitely correct in calling their installer "Universal". Every Oracle database product uses that installer, so we will become very familiar with it if we have to install any more Oracle products. It is universal also in the fact that it runs on every platform that Oracle supports, and so the same interface is used no matter where we install it. The installation steps are as follows:

1. The first step it goes through is asking us for the Oracle home details. It's similar to what it asked at the beginning of the database installation as shown in the following image:





The installer will again suggest **OraDb11g\_home1** or something similar, but we'll change it to **OraOWB11g\_home1** since it's just the OWB installation and not the full database.

When installing the standalone OWB software, remember that it cannot be installed into the same ORACLE home as the database. It must reside in its own Oracle home folder. So if we have a database that's already installed on the same machine, we'll have to make sure the ORACLE\_HOME we specify is different. The installer will warn us if we try to specify the same one and won't let us continue until it is different.

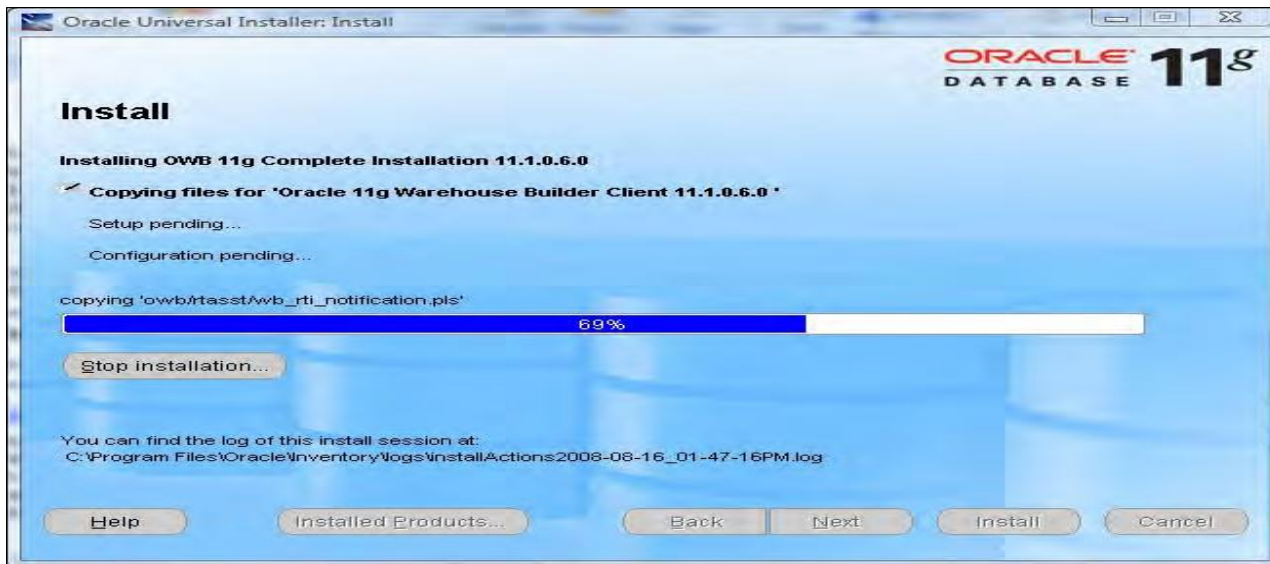
We need to verify the installation location for the home location also. The suggested name that it provides conforms to the OFA standard just as the database installation did, so we'll want to just change the drive letter if needed. However, the bottommost folder name can be changed if needed without violating the OFA standard. If it has a default of **db\_1**, we can change it to **OWB\_1** just to be clear that it's the OWB client.

2. The next and final step is the summary screen. The OWB client installation is not as complex as a full database installation, so it does not need all the additional information it asked for during the database installation. The summary screen should look similar to the following:



This summary gives us an idea of the disk space it will need, as well as the products that will be installed. If we scroll down the list, we'll see a number of other Oracle utilities and applications that it will install. We will also see items that are installed on the server as a part of the database install, but that will now be available to us on our client workstation. **SQL\*Plus** appears there, which is the command line utility for accessing an Oracle database directly using **SQL (Structured Query Language)**, the language used for accessing information stored in databases) among a host of other features.

Upon proceeding, the next screen will begin the installation and present us with the progress screen with a sliding bar moving to the right to indicate how far it has progressed. This is similar to what it did for the full database installation. An example of that screen is included next for reference:



### Install results

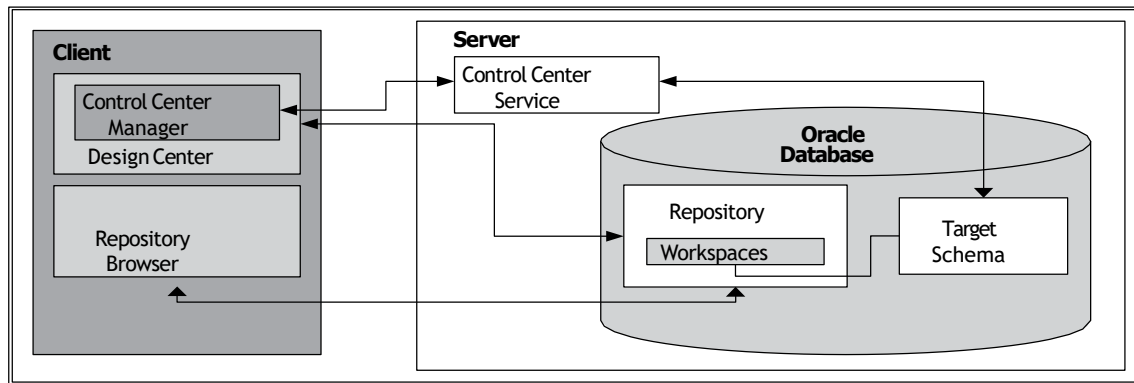
The log files with the results of the installation are stored in the same location as they are for a full database install. The universal installer will use that same folder for all its installs.

When the installation is complete, we will be presented with the final success screen and an **Exit** button. And as if to remind us about the universal nature of the installer, it will pop up a confirmation box asking if we really want to exit, even though for this installation there is nothing else that would be available to do on that final screen if we said no.

## OWB components and architecture

Now that we've installed the database and OWB client, let's talk about the various components that have just been installed that are a part of the OWB and the architecture of OWB in the database. Then we'll perform one final task that is required before using OWB to create our data warehouse.

Oracle Warehouse Builder is composed on the client of the **Design Center** (including the **Control Center Manager**) and the **Repository Browser**. The server components are the **Control Center Service**, the **Repository** (including **Workspaces**), and the **Target Schema**. A diagram illustrating the various components and their interactions follows:



### Client and server



The previous diagram depicts a client and server, but these are really just logical notions to indicate the purpose of the individual components and are not necessarily physically separate machines. The client components are installed with the database as we've seen previously and, therefore, can run on the same machine as the database. This configuration is assumed throughout the book.

The Design Center is the main client graphical interface for designing our data warehouse. This is where we will spend a good deal of time to define our sources and targets, and describe the **extract, transform, and load (ETL)** processes we use to load the target from the sources. The ETL procedures are what we will define to carry out the extraction of the data from our sources, any transformations needed on it and subsequent loading into the data warehouse. What we will create in the Design Center is a logical design only, not a physical implementation. This logical design will be stored behind the scenes in a Workspace in the Repository on the server. The user interacts with the Design Center, which stores all its work in a Repository Workspace.

We will use the Control Center Manager for managing the creation of that physical implementation by deploying the designs we've created into the Target Schema.

The process of deployment is OWB's method for creating physical objects from the logical definitions created using the Design Center. We'll then use the Control Center Manager to execute the design by running the code associated with the ETL that we've designed. The Control Center Manager interacts behind the scenes with the Control Center Service, which runs on the server as shown in the previous image. The user directly interacts with the Control Center Manager and the Design Center only.

The Target Schema is where OWB will deploy the objects to, and where the execution of the ETL processes that load our data warehouse will take place. It is the actual data warehouse schema that gets built. It contains the objects that were designed in the Design Center, as well as the ETL code to load those objects. The Target Schema is not an actual Warehouse Builder software component, but is a part of the Oracle Database. However, it will contain Warehouse Builder components such as synonyms that will allow the ETL mappings to access objects in the Repository.

The Repository is the schema that hosts the design **metadata** definitions we create for our sources, targets, and ETL processes. Metadata is basically data about data. We will be defining sources, targets, and ETL processes using the Design Center and the information about what we have defined (the metadata) is stored in the Repository.

The Repository is a Warehouse Builder software component for which a separate schema is created when the database is installed—OWBSYS. This is the schema we talked about unlocking during the installation discussion previously as one of the final steps in the database creation process. This will be created automatically by the 11g install, but is installed separately using scripts if we want to host the Repository on

an Oracle 10g database. The explanations in this book all assume that the Repository is hosted on an Oracle 11g database. The *Oracle Warehouse Builder Installation and Administration Guide* found at the following URL: [http://download.oracle.com/docs/cd/B28359\\_01/owb.111/b31280/toc.htm](http://download.oracle.com/docs/cd/B28359_01/owb.111/b31280/toc.htm) discusses the procedure for installing the Repository schema on an

Oracle 10g release 2 database if needed.

The Repository will contain one or more Workspaces as shown in the previous diagram. A Workspace is where we will do our work to create the data warehouse. There can be more than one workspace defined in the Repository. A common example of how multiple workspaces can be employed is to use different workspaces corresponding to sets of users working on related projects. We could have one workspace for development, one for testing, and one for production. The development team could be working in the development environment separately from the test team that would be working in the test environment. For our purposes at the ACME Toys and Gizmos Company, we will be working out of one workspace.

This concept of the workspace is new in this latest release of OWB. The Repository is created in the OWBSYS schema during the database installation. So setting up the Repository information and workspaces no longer requires **SYSDBA** privileges for the user to install the Repository. SYSDBA is an advanced administrative privilege that is assigned to a user in an Oracle database. This allows the user to perform tasks affecting the database and other database users that ordinary user accounts cannot do (or for that matter, other administrative accounts without SYSDBA). For security reasons, we want to restrict user accounts with SYSDBA privilege to a minimum. So it is good that we don't have to use that privilege when we install the Repository.

One final OWB component to consider is the Repository Browser on the client. It is a web browser interface for retrieving information from the Repository. It will allow us to view the metadata, create reports, and audit runtime operations. It is the only other component besides the Design Center and the Control Center Manager that the user interacts with directly.

We will have a chance to visit each one of these areas in much more detail as we progress through the design and build of our data warehouse. However, first there is one more installation step we have to take before we can begin using the Warehouse Builder. The Repository must be configured for use and a workspace must be defined.

## Configuring the repository and workspaces

We have talked about the OWBSYS schema that is created for us automatically during the Oracle 11g installation, and we have also looked at unlocking it and assigning a password to it. However, if we were to connect to the database right now as that user, we would find that as yet no objects exist in that schema. That is what will be done during this final installation step. We are going to use the **Repository Assistant** application to configure the repository, create a workspace, and create the objects in the repository that are needed for OWB to run. This application is available from the **Start** Menu under the **Warehouse Builder | Administration** submenu of the Oracle program group as shown here:



These menu options will appear locally on a client if we've installed the standalone Warehouse Builder client, as well as on the server. So where should we run the **Repository Assistant** if we have both? The most common configuration is to run this application on the same machine where the repository is located and the Control Center Service is going to run, which is all on one machine. There are other less common options for where to run the Control Center Service and where the Repository is located in relation to the target schema. These options are documented in *Oracle Warehouse Builder Installation and Administration Guide*. The URL for the chapter in the guide is the following: [http://download.oracle.com/docs/cd/B28359\\_01/owb.111/b31280/install\\_rep02.htm#BABJDHCJ](http://download.oracle.com/docs/cd/B28359_01/owb.111/b31280/install_rep02.htm#BABJDHCJ). This information can also be found in Chapter 2, in the subsection on implementing a remote runtime .

We want the runtime implemented on the server, which is the most common and simplest configuration. The Repository Assistant pops up an extra screen if it is running remotely from the client, which we will see next. We would see it during the installation if we were on a remote computer.

The steps for configuration are as follows:

1. We'll launch the **Repository Assistant** application on the server (the only machine we've installed it on) and the first step it is going to ask us for the database connection information—**Host Name**, **Port Number**, and **Oracle Service Name**—or a **Net Service Name** for a **SQL\*Net connection**. SQL\*Net is Oracle's networking capability for communicating with databases in a distributed networked environment. A naming method is configured so that when using a Net Service name, SQL\*Net will know what connection information to use for the connection. We have not configured a naming method, since we don't really need it just to connect locally, so we'll use the **Host Name**, **Port Number**, and **Oracle Service** name option as follows:
  - The **Host Name** is the name assigned to the computer on which we've installed the database, and we can just leave it at **LOCALHOST** since we're running it on the computer that has the database installed.
  - The **Port Number** is the one we assigned to the listener back when we had installed it. It defaults to the standard 1521. This is an example of why the issue of changing or not changing that default port number was mentioned. If we changed it but can't remember what we changed it to, then the following tip will help out.

#### Determining what port your listener is listening on

There are a couple of options we have for this. One is to perform the following steps:

Open a command prompt window and type in the following command:

```
C: \>lsnrctl
```



This will launch the **Listener Control** program, which is the command line utility Oracle provides for controlling the listener. Then enter the following command at the listener control prompt:

```
LSNRCTL> status
```

Look for the line that says:

**Listening Endpoints Summary...**

The next line will have the port number listed along with the protocol and host name such as the following:

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=computer)(PORT=1521)))
```

We can find information about the second option for determining the port number in the listener configuration file, `listener.ora`, in the Oracle home `NETWORK\ADMIN` directory. Open that file with Notepad and look for the above line.

- For the **Service Name**, we will enter the name we assigned to our database during step 3 of the database creation process. The name we used is `ACMEDW`. At the end of the database configuration assistant process, a detail screen was displayed. It was suggested that it would be a good thing to take a screen capture of it because it contained details about the database configuration, which would be useful later. One of the items on that screen was the database name that was assigned. If that is not available, then here's another tip to find the database name.

#### Finding your database instance name

There are a number of places where the database name appears on the database server without us having to log in to the database. One is in the listener control program. Open a command prompt window and type in the following command:

```
C:\>lsnrctl
```

This will launch the Listener Control program. Then enter the following command at the listener control prompt:



```
LSNRCTL> service
```

Look for the instance name in the list of services that appears.

Another option is to check the name of the Windows service that is started for the database. The database service name is a part of that name. Open **Control Panel | Administrative Tools | Services**. The Windows service names for the Oracle processes all start with `Oracle`. The service that runs the actual database is named `OracleService<dbname>` where `<dbname>` is the name of the database instance that you are looking for. The name says `OracleServiceACMEDW` for a database name of `ACMEDW`.

We can also check the Oracle base folder, which is the folder where the Oracle software was installed. The `Admin` folder contains a folder named for the database instance if we followed the default naming conventions for folder names during the installation. That is one reason to stick with the OFA standard when installing Oracle products.

2. Now that we've determined the connection information for our database, we'll move along to step 2 of **Repository Assistant**. It asks us what option we'd like to perform of the following:
  - **Manage Warehouse Builder workspaces**
  - **Manage Warehouse Builder workspace users**
  - **Add display languages to repository**
  - **Register a Real Application Cluster instance**



We're going to select the first option to manage workspaces and move along to the next step.

3. This step asks us what we'd like to do with workspaces: create a new workspace or drop an existing one. We'll select the first option to create a new workspace.
4. This brings us to step 4 of the process, which is to specify an owner for the workspace. We are presented with two options: to create a new user or to be the owner. To perform the first option, we will need to specify a database user who has DBA privileges that are required to be able to create a new user in the database. The second option is to specify an existing database user to become the owner of the workspace. This user must have the **OWB\_USER** role assigned to be able to successfully designate it as a workspace owner. That is a database role required of any user who is to use the Warehouse Builder. If the existing user who is selected does not have that role, then it must be assigned to the user. An additional step will be required to specify another user who has the ability to do that assignment (grant that role) or has DBA privileges. This second user must have the **Admin Option** specified for the OWB\_USER role to be able to grant it if he or she does not have DBA privileges.

The user specified here, whether new or existing, will become a deployment target for the Warehouse Builder. This means that the user will be able to access the Design Center for building the ETL processes and the Control Center Manager for deploying and auditing. We'll specify a new user for the ACME Toys and Gizmo's warehouse, since we've just installed this database and no other users are created yet.

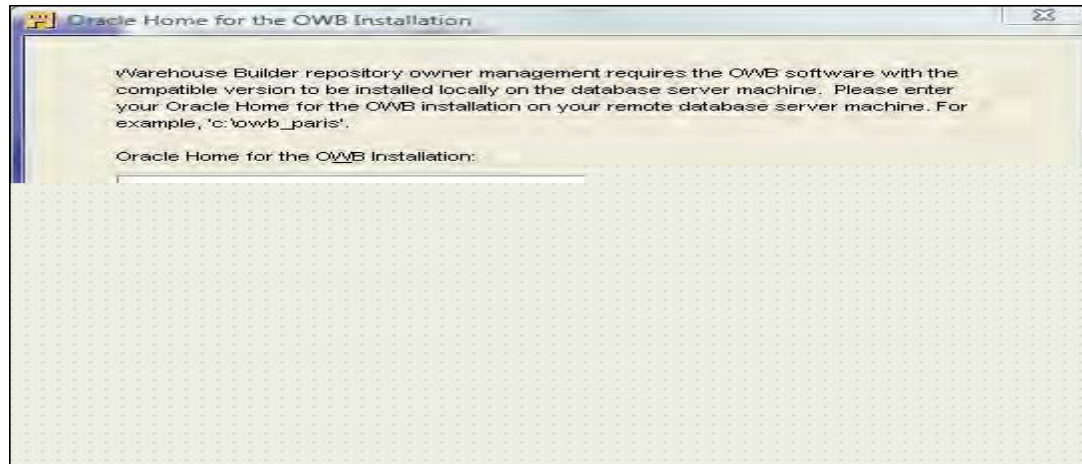
5. This step will depend on which option we specified in step 4. If we are creating a new user, it will ask us for an existing user with DBA privileges in the database. The SYSTEM account is the default provided there, but if we have a different account that is a DBA in the database, we can use that. If we have specified an existing user in step 4, then step 5 will ask us for the username and password for that user, as well as the name of the new workspace to create.

Since we're specifying a new user, we will put in the password for the system user and proceed to the next step. The password used here is the one we previously defined for the system accounts when we created our database.

6. In this step we specify the new username, password, and workspace name. We'll use **acmeowb** for the username and **acme\_ws** for the workspace name.

7. This step will ask for the password for the OWBSYS user. This schema was installed for OWB to use for the repository. The password it's looking for is the one we set up back on the final database configuration screen at the end of running **Database Configuration Assistant** to configure the database. This step will only be required upon first running **Repository Assistant** to create a new workspace since it also has to perform the process of initializing the repository in the OWBSYS schema first. That is a one-time process which is why subsequent runs of **Repository Assistant** to manage workspaces, will not require this step.

After putting in that password, if we were running the **Repository Assistant** on a different machine than the database was installed on, then we would encounter the following screen. We referred to it earlier when talking about running the **Repository Assistant** remotely.



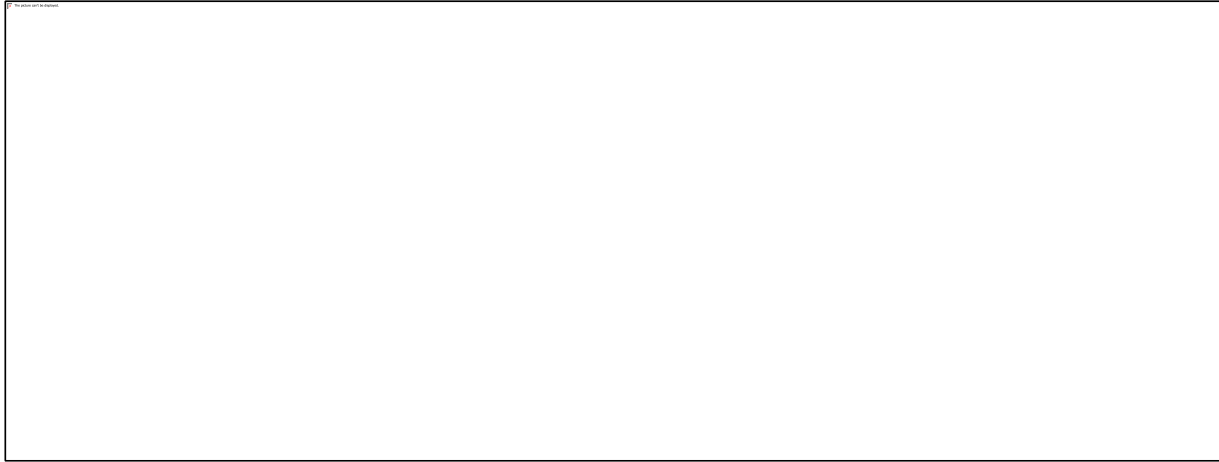
It doesn't know the location of the Oracle Home on the server, and so must prompt for it. It also provides the option for a **Local Control Center Service** that is for the remote runtime option discussed in the installation guide. Since we're running our database on the same machine as our client, we won't see this screen.

8. This step asks for **tablespace** names for the OWBSYS schema. A tablespace is a logical entity in an Oracle database for storing data. All objects created are assigned to a tablespace, which stores the data physically in a datafile or datafiles assigned to the tablespace. The administration of tablespaces in an Oracle database is more than we have room for here, so we won't be creating any new tablespaces to hold the OWBSYS data. We'll just leave the defaults selected—the **USER** tablespace for data indexes and snapshots, and the **TEMP** tablespace for temporary data. For advanced production databases, it would be a good idea (at a minimum) to specify a separate tablespace for OWBSYS, and actually think about using three new tablespaces for those three that have the USER tablespace assigned.
9. This step is to select a base language for the repository, so we'll make the appropriate selection. Once the repository is created, we cannot change the base language and there can only be one base language assigned to the repository. Physical names of repository objects are assumed to be in the base language. The **Repository Assistant** will automatically assign the base language depending on the locale that is assigned to the computer we're installing on. We also have the option of selecting one or more display languages that will allow users to assign a business name to physical objects in their own language. Unlike the base language, we can assign display languages after the repository is created. Select any of those that apply.
10. We're almost finished. The final step is the optional step 10 to specify any workspace users from existing database users. We specified the workspace owner as a new user earlier in the install process, and now it's asking for any additional users who we might want to have access to the workspace. The workspace owner is allowed to add and remove database users from the workspace.



Removing a database user from the workspace does not delete that user account from Oracle. It only removes him or her as a valid user of the workspace. ]

After selecting any user, the **Repository Assistant** will present us with a summary screen of the actions it will take and the information we entered, as shown in the following image:



Notice the name of the project at the end. There was no option to specify that project name, so it's just using a default name. It always sets up a default project in a new workspace by that name, but we can change it later when we actually start designing our data warehouse and working with the workspace in the Design Center.

Click on the **Finish** button and it will begin the installation, presenting us with a scroll bar moving to the right as it progresses through the installation. The very first time it runs, it will take around 5 to 10 minutes to run before reporting the success pop up, as it has to initialize the repository in the OWBSYS schema. Creating new workspaces after the first time will be very quick, taking no more than a few seconds to complete.