

An Oracle White Paper  
May 2010

# Oracle SQL Developer Migration

Overview .....	3
Introduction .....	3
Oracle SQL Developer: Architecture and Supported Platforms.....	3
Supported Platforms .....	4
Supported Databases .....	4
Connections .....	4
The Migration Process.....	5
Standard Migrate .....	5
Quick Migrate .....	6
Online or Offline Modes .....	6
Capturing the Non-Oracle Database.....	7
Migrating the Data .....	7
Planning the Migration.....	7
Resources .....	10
Conclusion .....	10

## Overview

Migrating from one database to another is often a high-risk, expensive, and time-consuming process. However, Oracle provides products that reduce the time, risk, and financial barriers involved in migrating non-Oracle databases to the Oracle platform.

Oracle SQL Developer provides an integrated migration tool for migrating Microsoft SQL Server, Sybase, MySQL, Microsoft Access, IBM DB2 LUW and Teradata to Oracle Database.

The initial production release of Oracle SQL Developer was in March 2006, with the migration capabilities added to SQL Developer 1.2, released, June 2007. The latest release is Oracle SQL Developer 2.1.1, available from March 2010. This white paper outlines the migration capabilities of Oracle SQL Developer.

## Introduction

SQL Developer is a graphical tool that enhances productivity and simplifies database development tasks. Using SQL Developer, users can browse, create and modify database objects, run SQL statements, edit and debug PL/SQL and have access to an extensive list of predefined reports or create their own.

With SQL Developer, users can create connections to non-Oracle databases including Sybase, MySQL, Microsoft SQL Server and IBM DB2 for object and data browsing. Once a connection is created, the tool provides utilities to migrate any of these databases to Oracle. Depending on the database in question, SQL Developer automatically converts the tables, triggers, stored procedures and all other relevant objects to an Oracle database. Once the target Oracle database has been generated, SQL Developer assists in the migration of the data from the non-Oracle database to the target Oracle database.

## Oracle SQL Developer: Architecture and Supported Platforms

Oracle SQL Developer is developed in Java and sits on the Fusion Client Platform (FCP). There are different distributions of SQL Developer available, depending on the platform required. For Windows clients, users can elect to download the version bundled with the JRE 1.6. This has an additional tools.jar to support the Windows clients. For Windows clients not using the bundled distribution, and

all other platforms, users should download the zip file and direct SQL Developer to access a separately available or installed JDK 1.6.

## Supported Platforms

Oracle SQL Developer 2.1.x is certified to run against JDK 1.6, update 6 or above. Platform Support is as follows:

PLATFORM	SUPPORTED VERSION
Microsoft Windows	Windows 2000 Windows 2008 Server XP (Professional) Vista
Apple MAC	OS X (Power PC)
Linux	Oracle Enterprise Linux 4 Red Hat Enterprise Linux SUSE Linux Enterprise Server

Figure 1: Supported Platforms

## Supported Databases

SQL Developer currently supports migrations from non-Oracle databases to Oracle Database 10g and Oracle Database 11g. The following non-Oracle databases are supported:

SUPPORTED NON-ORACLE DATABASES	VERSION
Microsoft SQL Server	7.0, 2000 and 2005
Microsoft Access	97, 2000, 2002 and 2003
Sybase Adaptive Server	12 and 15
MySQL	3, 4 and 5.0
IBM DB2	LUW 7, 8 and 9
Teradata	12

Figure 2: Supported Non-Oracle Databases

## Connections

The migration capabilities are an integral part of SQL Developer, and are available by default, when you install SQL Developer. Before starting a migration, you need:

- A Connection to an Oracle database.

- A Connection to your non-Oracle database, or offline capture-files.  
Creating connections to an Oracle and Microsoft Access database is built into SQL Developer by default. Creating connections to Microsoft SQL Server, Sybase, IBM DB2 LUW, Teradata and MySQL require database-specific JDBC drivers.
- A migration repository  
SQL Developer's migration tools require a migration repository to persist information during the migration. Once the migration repository is associated with an Oracle connection, SQL Developer will automatically define the tables and packages required by the migration tools.

## The Migration Process

A migration, like any other software project, requires analysis, planning and testing. It should be noted that SQL Developer does not provide tooling in those areas; instead it has focused on automating as much of the database migration as possible.

SQL Developer provides two approaches for migrating non-Oracle databases:

- Standard Migrate
- Quick Migrate

### Standard Migrate

SQL Developer breaks down the migration process into four distinct phases.

#### Capture

The first step performed in a migration is to capture a snapshot of the non-Oracle database. This non-intrusive step gathers meta-data about the non-Oracle database. The information is saved as a captured model in the repository. This captured model can then be modified without any repercussions, for the non-Oracle database. Note that no table data has been captured as this is handled in the last step.

#### Convert

The captured model is then converted to an Oracle model. This model represents what the Oracle database would look like. It too can be modified without any repercussions, as it exists as only a model within the repository and not an actual database.

#### Generate

The converted model is then expressed as a SQL script, which when run against an Oracle database, will generate all the migrated users, tables, triggers procedures and other relevant objects.

#### Data Move

The last step is to migrate or move the data, which can be done online using the SQL Developer connections or offline, with scripts automatically created by SQL Developer. These scripts utilize the non-Oracle data move tools like Bulk Copy Program (BCP) and Oracle's SQL\*Loader.

Throughout this process SQL Developer provides a user interface to browse and modify the live non-Oracle database, the captured and converted models and the live target Oracle database. Figure 2 displays the Captured and Converted Models windows.

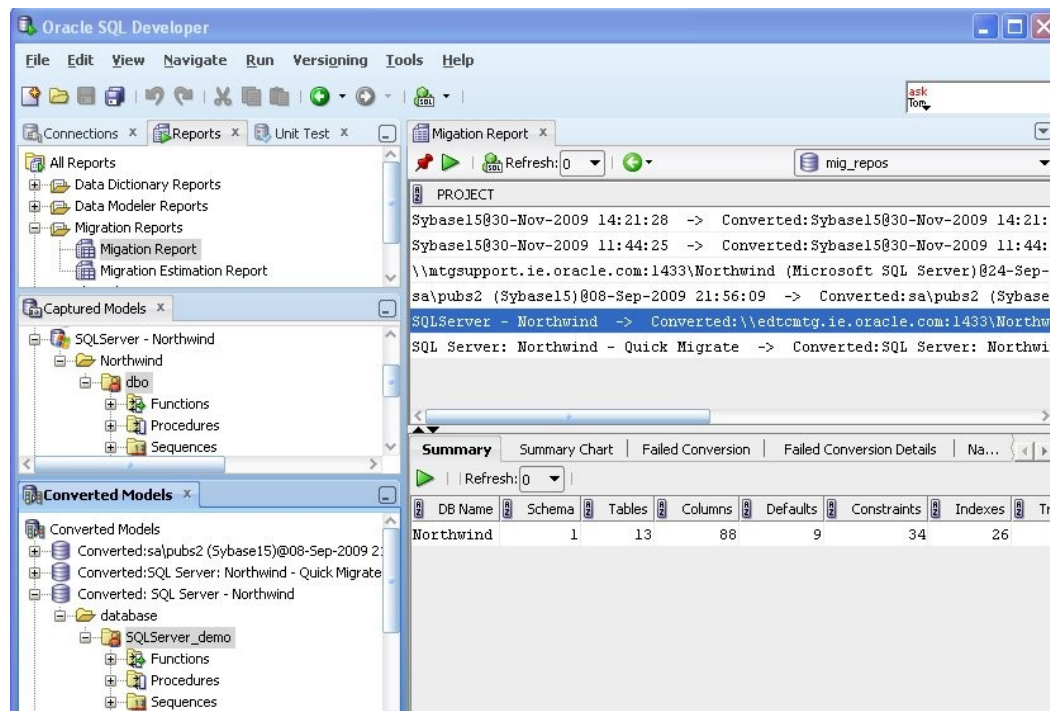


Figure 3: Capture and Converted Models

## Quick Migrate

The Quick Migrate option is essentially a one-step approach, where users step through a wizard, populating details such as the non-Oracle connection, target Oracle connection, and the information for the migration repository. It then takes the default settings and performs the steps described above: Capture, Convert, Generate and Data Move. This approach is only suitable for small simple databases or as a prototype exercise. Any issues that occur during the migration will force the Quick Migrate to stop.

The standard migration approach is recommended, because it allows intervention after each phase where issues can be resolved. It also provides more granular control of modifying structures, deleting and renaming objects.

## Online or Offline Modes

You can use different methods to capture and migrate the objects and data from the non-Oracle database. Broadly speaking these migrations are:

- Online, using a direct access to the non-Oracle database through a connection in SQL Developer or
- Offline, where a set of scripts and the corresponding output is used.

## Capturing the Non-Oracle Database

- Using the online capture, SQL Developer connects to the non-Oracle database through a live JDBC connection, to extract the database structure, and populate the captured model. Right-click on the non-Oracle connection and select *Capture...*, to start the capture process.
- SQL Developer provides a set of scripts to extract the meta-data from the non-Oracle database and load it into a set of files. SQL Developer then extracts the meta-data from the files into the migration repository. This method provides increased flexibility because you can perform a migration without having to install SQL Developer at a customer site. For example, a consultant can ship the extraction scripts to the customer who can run the scripts and return the output back to the consultant. The consultant can then perform the migration in the office, and then generate the required Oracle scripts.

## Migrating the Data

Once you have captured, converted and generated the target Oracle database there is once again a choice of how to complete the data move:

- Transfer the data online.  
This method is most suited to novice users and users with small amounts of data.
- Transfer the data offline.  
In this case; SQL Developer creates data migration scripts that migrate the data offline. This is the recommended method for databases with large amounts of data. To transfer the data offline, right-click on the Converted Model and use the Generate Offline Data Move Scripts.

## Planning the Migration

When undertaking any project, it is important to plan the project and identify the risks involved in completing the project on time and to budget. You can use SQL Developer during the planning and analysis phases of a migration project to identify the amount of work involved in the database migration.

The following steps outline the recommended planning process for your migration project:

### 1. Determine the Project Requirements

Define the scope of the project. You can do this by determining whether you have a simple or complex migration scenario. You may be dealing with a complex migration scenario if you answer yes to some or all of the following questions:

- Is your database greater than 25MB?
- Is your database a data warehouse?
- Does your database support large applications, that is, greater than 100 screens, forms, reports, and batch jobs?
- Do multiple lines of business use your database?
- Are you deploying your database in a distributed environment?
- Does your database support a large user base, that is, more than 100 users?
- Is high availability of your database a requirement, for example, 24 x 7 x 365?

The more complex your migration scenario, the longer your migration will take to complete. This is due primarily to the increased list of tasks associated with a complex migration. There is also a higher risk of problems occurring during the project. At this stage, you also define testing and acceptance criteria and create a requirements document with a list of requirements for the migration project.

## **2. Estimate the workload using SQL Developer**

SQL Developer is a useful scoping tool that you can use to estimate the workload involved in the migration. You do this by using the SQL Developer to capture the non-Oracle database and create the converted model. The SQL Developer Migration Reports can assist in evaluating the results of the capture and convert phase. The reports categorize the total number of database objects and allow you to identify the number of objects that can be converted and migrated automatically. They display the pass, fail and limitations and can offer an idea of how much of the migration you can perform automatically, helping estimate the amount of manual work involved.

## **3. Analyze the operational requirements**

You must analyze the operational requirements. You can do this by answering the following questions:

- What backup and recovery changes do you require?
- What downtime is required during the migration?
- What effect does the downtime have on the business?
- What are the performance requirements?
- Are you changing the operational time window?
- What training requirements or additional staff considerations are required?
- Is it necessary to have the non-Oracle database and the Oracle database running simultaneously?



#### 4. Analyze the application

You need to analyze the application from both a business and a technical perspective. You can do this by answering the following questions:

- Does the current application still meet the business requirements or do you need to re engineer it?
- Do you need to change the database connection statements or SQL statements to make the application run effectively on the Oracle database?

If changes are required to the application, determine whether it is more efficient to rewrite or modify the application. Allocate time and resources to address each issue associated with rewriting or modifying the application. Update the requirements document created for the project in Step 1.

SQL Developer provides a useful feature called the Migration Scratch Editor. This can be used to help migrate non-Oracle SQL within your application code.

#### 5. Plan the migration project

Using the budget constraints of the project and the information gathered from steps 1-4, produce a migration plan and estimate how much time the migration project should take. Then calculate how many resources are required to complete and test the migration.

#### 6. Perform the migration

SQL Developer automates the large majority of this step. For more information please download one of our 'Oracle by Example' tutorials available on the Oracle Technology Network.

#### 7. Finalize the migration

Although SQL Developer provides much automation, you may have a certain amount of manual work in order to complete the migration. The majority of this manual effort is required for SQL programs, such as triggers and stored procedures.

The Translation Scratch Editor can be useful at this stage by providing the translation of a SQL program side-by-side.

#### 8. Test the Oracle Database

After you have completed the migration, it is important to test the Oracle database and related applications. Testing ensures that the migrated data is accurate and complete. You should also ensure that the applications function in the same way as they did with the non-Oracle database.

#### 9. Deploy the Oracle Database

Deploying your newly migrated database and application within your business environment may be difficult. Therefore, it is important to employ the most suitable rollout strategy. For example, you may use a phased approach where you migrate groups of users at different times. Alternatively, you may use the big bang approach where you migrate all users at the same time. Another option is to use the

parallel approach where you maintain both the non-Oracle database and the newly migrated Oracle database simultaneously for a certain amount of time.

## Resources

The Oracle Technology Network (OTN) is the definitive source for Oracle technical information about developing on the Oracle platform. As a member of OTN, you are part of an online community with access to free software, OTN-sponsored Internet developer conferences, and discussion groups on up-to-date Oracle technology.

### Migration Center

Find a host of migration-related information from the Migration Center Web site at:  
<http://otn.oracle.com/tech/migration>

### Download Software

You can download Oracle SQL Developer free of charge from OTN at:  
<http://www.oracle.com/technology/software/products/sql/index.html>

### Documentation

Online documentation is available from the Oracle Technology Network Web site at  
[http://download.oracle.com/docs/cd/E15846\\_01/index.htm](http://download.oracle.com/docs/cd/E15846_01/index.htm)

### Technical Support

For technical support please refer to Oracle Support using Metalink.

Access the discussion forum on OTN at: <http://www.oracle.com/forums/forum.jsp?forum=260>

### Further Enquiries

Address all product enquiries about the Migration Workbench to the Migration Workbench helpdesk at: [mtg-feedback.ww@oracle.com](mailto:mtg-feedback.ww@oracle.com)

## Conclusion

The Oracle SQL Developer supports the migration of Microsoft Access, Microsoft SQL Server, MySQL, and Sybase Adaptive Server databases to Oracle. It reduces the migration time by performing a large portion of the migration automatically. It also reduces the risks involved by generating reports that enable you to estimate the manual effort required to complete the migration. Integrated into Oracle SQL Developer, the migration tool is available free of charge. The combination of all these factors shows SQL Developer to be the strongest solution in the market today for migrating non-Oracle databases to Oracle.



Oracle SQL Developer Migration  
May 2010  
Author: Sue Harper  
Contributing Authors: Dermot O' Neil, Barry  
McGillan

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

**SOFTWARE. HARDWARE. COMPLETE.**