

# Using Data Replication to Upgrade Your Oracle® Database with Minimal Downtime

Written By Bill Brunt, Product Manager, Dell Software



## Abstract

In these tough economic times, organizations are trying to lower costs in all of their business operations. One key method for cutting IT costs is migration: migration to hardware that is less expensive to purchase and operate, migration to less costly operating systems, and migration away from older versions of Oracle® that may require expensive extended-support contracts.

This paper provides an overview of the types of migrations and their challenges, and describes the established best practices for ensuring success. It also provides details on a proven process for upgrading to Oracle12c that minimizes downtime and the risk of failure by using a data replication solution such as Dell™ SharePlex™.

## Introduction

Migrations can be an effective way to reduce costs. Organizations can choose from several different kinds of migrations, depending on their needs:

- **Hardware migrations**—Many organizations with large Unix servers are looking to reduce operating costs, such as power, cooling, hardware and space requirements. These organizations often choose to migrate to newer, lower-cost hardware, such as servers or disk arrays, that are faster and more reliable than their old systems, or that can be configured as a RAC environment for the same or lower cost. Other options include migrating to a virtualized environment and run more VM servers on fewer large machines, or moving systems and databases to the cloud.
- **Operating system (OS) migrations**—Organizations are also reducing operating costs by moving applications onto less expensive operating systems, such as Linux. Operating system migrations can also be necessitated by a move to a virtualized environment. In addition to saving money, operating system migrations can offer valuable flexibility, such as the ability to allocate more memory to particular Oracle instances.
- **Oracle migrations**—A number of organizations are still running older versions of Oracle that either require extended support contracts or are no longer supported. Migrating to a current version of Oracle can reduce support costs, ensure the benefits of running a supported

The key challenges of migrations relate to their potential to disrupt the business.

version, and enable the organization to avoid spending capital on an extended support contract. Oracle Standard Edition, with its lower licensing costs, is becoming a more viable option as well. Hardware technology improvements are erasing its old resource and performance constraints—with chipsets having eight cores and now pushing 16 cores, running a database on four sockets no longer has a computing restriction. Moving to Oracle Standard Edition in place of Oracle Enterprise Edition can also reduce operating costs—Standard Edition is not as feature rich as Enterprise Edition, but it can run many departmental applications and it offers the same level of reliability and robustness at up to 40 percent less cost.

### Challenges of migrations

The key challenges of migrations relate to their potential to disrupt the business. Therefore, organizations need strategies to minimize user downtime, testing applications to ensure that they will continue to work properly after the migration, and planning for recovery to the old environment if issues arise.

#### Minimizing downtime

Before the era of 24x7 operations, upgrades were scheduled over the weekend. Users would log off early Friday afternoon, IT staff would perform a complete backup, and then the upgrade would start. The main goal was to make sure the systems were operational Monday morning—either because the upgrade was successful or because the backup from Friday afternoon had been restored before users arrived on Monday morning.

Today, being offline for an hour, much less a whole weekend, is completely unacceptable, especially for environments such as e-commerce, where a major portion of the week's revenue is generated on weekends. Therefore, every step of the upgrade process must be fine-tuned to minimize downtime for the end user. Is a cold backup necessary? Can we just make a disk copy of the database? Do we have

the additional space needed for the migration? Do we have the best recovery technique?

Some risks can be anticipated and minimized. For instance, an upgrade requires more disk space and memory than is used in day-to-day operations; if the system doesn't have these extra resources available, the upgrade could easily fail. Other failures can be the result of unexpected problems, such as media failure, a bad backup or a power outage. Some of those risks can be reduced by checking the validity of a backup, which takes a few minutes but is well worth the investment. Imagine needing to revert to the backup and finding that your fallback process is flawed!

#### Application testing

Another significant migration challenge is adequately testing applications after migration to avoid the need for rolling back to the pre-migration environment. Ideally, after the migration, you want to thoroughly test your applications before making them available to your user community, but this is seldom possible due to the pressure to get the system back online as fast as possible. Most organizations try to save downtime by limiting the application testing—which increases the risk of a failed migration.

#### Planning for recovery

Organizations need to plan for the possibility that the migration is unsuccessful. What happens if an important application does not perform as well on the new version of Oracle or the new OS with a full production workload?

Even worse, what if the migration works flawlessly and your users work with the application for a couple of days, but then some part of the application does not work correctly? You must plan a way to recover back to the old version of Oracle or the old OS where everything was working, without losing the transactions from the last few days.

## Best practices for migrations

Following the best practices outlined below can help ensure the success of your hardware, operating system or Oracle migration by minimizing the risks of downtime and need for rollback.

### Plan carefully.

There are risks inherent in any migration, including media failures, power outages, and running out of time, memory or disk space. Proper planning can help mitigate these risks. Be sure to test as many steps of your migration process as you can—and to test multiple times. Document the process carefully to minimize the risk of human error: instead of having to remember various details, you can simply follow your written plan.

### Consider using export/import to restructure data.

If you are migrating to a new operating system, the only way to move your data to the new OS is by executing an export/import; you cannot simply back up the database and recover to the new server because of file format incompatibilities. But even migrations that do not require export/import can often benefit from it. Export/import enables you to restructure all of the data in your database and clean up any fragmentation. Laying out a database differently can relieve I/O hot spots or enable you to spread out the data into separate tablespaces. However, be aware that with large databases, the export/import process can take a long time.

### Disable triggers and cascade delete constraints.

In any migration, when using a replication technology such as SharePlex, be sure to disable triggers and cascade delete constraints on the new instance until it is time to go live in that environment. Otherwise, if the triggers and constraints were to manipulate data in the old environment, those data changes will be replicated to the new instance. If the triggers or cascade delete constraints are enabled on the

target, they will execute again and will cause out-of-sync conditions.

### Import into a new database shell.

To take advantage of some of the features of recent Oracle versions, you need to import into a new database shell rather than use the database upgrade script. For example, version 9i included the ability to change the block size of tablespaces, which is achieved by recreating the tablespaces. If you did not take advantage of these new features when you migrated to 9i, or your application did not support them, be sure to import into a new database shell so you can build the tablespaces in order to use this functionality.

### Plan your rollback strategy.

Be sure to plan a rollback strategy in case something does not work correctly in the migration. If you are performing an OS upgrade or Oracle migration to a new server, you can usually recover your backup of the original database to the new server. In the case of an OS upgrade, you don't need to do anything else. In the case of an Oracle upgrade, you also need to go through the Oracle upgrade process after the recovery is complete.

A safe failback option is to have a replication solution such as SharePlex. With SharePlex, you can replicate from the new version of Oracle back to the older version. This safeguards your data if your application does not work correctly after migration, regardless of how much work your users have done since the migration. You can failback to the older version with no loss of user data.

### Consider investing in a log-based replication solution.

A log-based replication solution such as SharePlex will help minimize downtime and the impact of the migration on your business. With such a solution, you can manage the migration process behind the scenes, and you'll never have to take your users offline except for a short

A log-based replication solution such as SharePlex will minimize downtime and the impact of the migration on your business.

period when you switch them over to the new database. This cuts down the impact on users during the upgrade. The length of the migration probably is not any shorter, but with the right replication technology in place, you will be able to mitigate risks such as downtime, application testing, platform validation and storage validation.

### The Oracle migration process

You can help ensure a successful Oracle upgrade and minimal interruption to production by using the right procedure and taking advantage of data replication technology.

The procedure we recommend for migrating from older database versions of Oracle to Oracle 12c involves the following steps. (We use Oracle 11g in our example, but this procedure can apply for any older Oracle version.)

1. Set up replication from source (Oracle 11g) to target (Oracle 12c).
2. Create a target instance (Oracle 12c).
3. Upgrade the target instance.
4. Resume replication from source to target.
5. Test the target instance thoroughly from platform validation to application validation.
6. Set up replication from target back to source.
7. Switch users to the target (use the source instance as a fallback if necessary).
8. Upgrade the source instance.

### Step 1. Set up replication from source (Oracle 11g) to target (Oracle 12c).

First, you will need to set up data replication between source and target so they can continuously replicate production data. The reason that we start replication before creating the new database is that we want some overlapping data in order to ensure we don't miss any data. Some of the tests to validate the target's performance require a change in data volume. As replication propagates data to the target instance, be sure to stress-test to make sure that it will be viable when you switch the users over. When you initiate the data replication, make sure that you are replicating all of the production tables and sequences so your tests are complete.

If you are using a log-based replication product like SharePlex, be certain to disable triggers on the target instance. Triggers should not modify the target data, since the results of their actions on the source system were recorded in the redo logs and already replicated.

### Step 2. Create a target instance (Oracle 12c).

Next we proceed with the copy of the production database on which to practice and test the upgrade. This can be accomplished in many ways, physical and logical, but the easiest method is to use disk mirroring: split the mirrors. Mount the copy on the same mount point on another system and open

Ensure a successful Oracle upgrade and minimal interruption to production by using the right procedure and taking advantage of data replication technology.

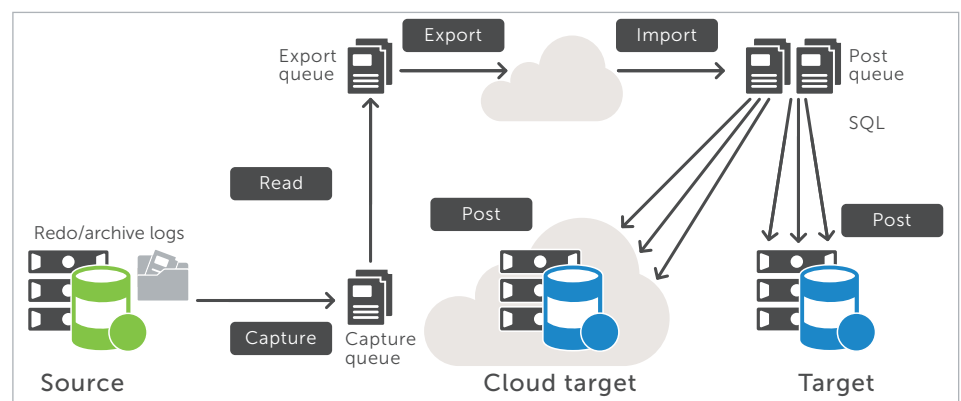


Figure 1: SharePlex data replication architecture

the instance. If you need to create the database logically, proceed as explained in Step 3.

### Step 3. Upgrade the target instance.

Once you have the target database created, you are ready to perform the upgrade on the target instance. Perform the upgrade per Oracle's instructions.

If logical instantiation is needed due to platform differences or restructuring of the new target database, create an empty Oracle 12c database and use datapump export and import to load the data to the target database.

### Step 4. Resume replication from source to target.

When you are finished with the upgrade of the target instance, remove duplicate data from replication by a reconciliation process that matches the recovery database with replication so you don't have any mismatches. Once this is done, allow the queued transactions to be applied to your new Oracle 12c target instance.

### Step 5. Thoroughly test the target instance.

#### Step 5a. Perform preliminary checks.

Replicating production activity is a great way to start testing your newly upgraded instance, because it provides a volume and variety of transactions that would take hours of development work to duplicate. Use replication for at least two days as the only form of test against the instance. Next, begin running read-only tests, checking your reports and queries for Oracle 12c compatibility.

#### Step 5b. Implement desired Oracle 12c features.

Implement the Oracle 12c features that influenced your decision to upgrade to Oracle 12c.

#### Step 5c. Test updating applications on Oracle 12c.

This is undoubtedly the largest part of the task—verifying applications, packaged and custom, are compatible

with the new database. This consists of multiple steps:

1. **Plan the tests**—First, make a list of all the applications you will test. You may want to organize the list by importance or complexity. (You also may need to organize some of the applications in respect to interdependencies.) The order of your list is not as important as its completeness. Having a list that you can update with the date, time and results is important, because you may forget which applications you actually tested. Be sure to include special periodic applications, such as month-end, quarter-end and year-end processes.
2. **Run a test**—To test an application that updates the data in your target instance, first stop the replication update process. Next, start a development diagnostic tool so you can determine which tables and sequences are affected by the application. Then run the test and check its results.
3. **Reset the test environment after each test**—Your options for resetting your target instance depend on the technologies available to the project and the extent of changes made by your tests. If the application you tested made a few changes to some large tables, a quick method to re-sync them is to use DataEquator, a compare-and-repair utility provided with SharePlex. It compares the data within pairs of tables and brings the anomaly back into synchronization. Another option is to simply store the target database to tape prior to the test and restore it afterward. Alternatively, use the flashback mechanism in Oracle to roll back the tests.

Another option is to re-sync from the production system. To do this, you'll need a list of all the schema changes made on the target instance, both from the upgrade itself and from the Oracle 12c features you implemented. Dell™ Toad™ for Oracle facilitates this process by comparing schemas on the two systems and creating a script of deltas. Don't run the script on the production system; run it on a copy of the production system. With the script prepared, repeat the process you performed to create the target instance: break the mirrors on the production system, move the copy to the target system, mount

Replicating production activity is a great way to start testing your newly upgraded instance.

it on the same mount point, and open the instance. Next, reconcile the image created using the mirroring technique with the transactions contained in the replication queues, eliminating transactions already reflected in the image created through mirroring. Finally, run the script to apply the schema changes, and you'll have built a fresh, consistent image of production data on your Oracle 12c target instance. This method also allows you to test your upgrade process as many times as you like.

4. **Re-sync the target instance**—When you have completed your test cycles, re-sync the target instance with the production instance once more.

#### **Step 6. Set up replication from target to source.**

Next, set up replication in the opposite direction: from Oracle 12c to Oracle 11g. Be careful to avoid setting up a never-ending loop for any transaction. The default mode of SharePlex disables such looping of replication. If you are using another replication product, verify that it will not replicate from one system to the other and back again, endlessly.

#### **Step 7. Switch users to the target instance (Oracle 12c).**

When the production system is at its least active state, quiesce the production system and flush all transactions to the Oracle 12c target instance. Once all the transactions have been posted, allow the users to begin working on the new Oracle 12c system.

#### **Use the source instance as a fallback.**

There's nothing like having users on a new version of software and the database to help you identify things missed during testing. All issues should be minor. However, if a major issue does arise at a critical time, you have a fallback plan in place: since you are now replicating from Oracle 12c to Oracle 11g, you have the option of redirecting your users to the source 11g system, which is now serving as a backup. The backup system has all of the users' work on an Oracle 11g instance, where your applications ran successfully

in the past. If a critical application hits a significant snag with Oracle 12c, you can switch the users back to Oracle 11g database and continue business. The IT staff can then determine what needs to be modified to make the application fully compatible with Oracle 12c, fix it, and prepare to switch the users to the Oracle 12c database again.

#### **Step 8. Upgrade the original source 11g system in minutes.**

Once your production environment has run successfully with the Oracle 12c database and applications long enough to give you confidence in the new version, you will want to upgrade your backup system to Oracle 12c. This process is faster and easier than the original upgrade because you have the data and the applications ready. You simply need to make a copy for the other system.

Repeat the process you performed when you created the test instance, but in reverse. Split the mirrors on the Oracle 11g system, move the copy to the backup system, mount it at the same mountpoint, and open the instance. Then tell the replication process to perform its reconciliation so that it eliminates transactions from its queues represented in the instance created via the mirroring solution, and then resume active replication.

Before shifting the users to the original production system (which has been serving as the backup), set up replication in the opposite direction (source to target). This will prepare you to resume normal operations. When ready, simply quiesce the production system (the target instance), flush the transactions to the backup system (the source), and redirect the users.

With replication in place, you can use the secondary instance for day-to-day reports, queries or high availability needs—maximizing your primary system's OLTP performance and availability. In addition, you can

If a critical application hits a significant snag with Oracle 12c, you can switch the users back to Oracle 11g database and continue business.

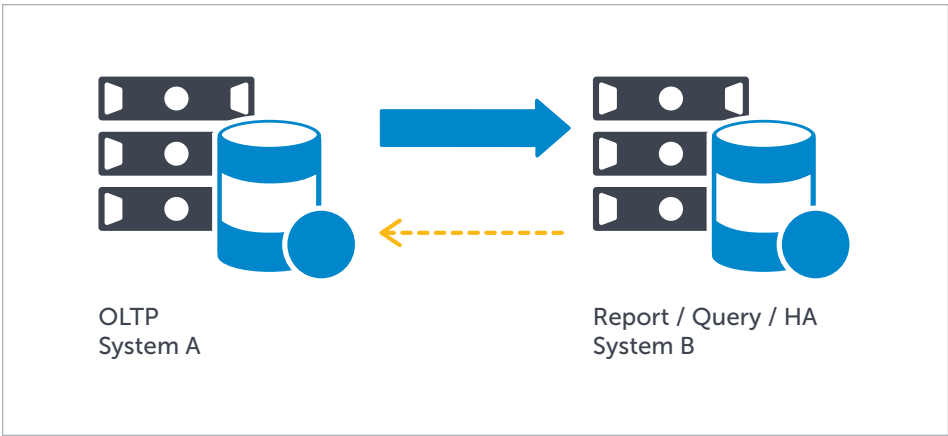


Figure 2: Optimize OLTP performance by leveraging replication for reporting, querying and high availability.

perform maintenance on the secondary system, while activity continues on the production system.

**Conclusion**

A hardware, operating system or Oracle migration can deliver significant cost savings. You can help ensure your migration is successful and minimize disruption to the business by following established best practices and investing in a data replication solution such as SharePlex. SharePlex enables you to manage the movement of data to the new environment while your application is still up and running, making the migration seamless to your users.

In fact, SharePlex customers using the migration procedure outlined in this paper have reduced their downtime by an average of 94 percent: the interruption to production users is limited to the time required to switch them from one system to the other and time for replication activation. This approach also

minimizes risk, because the upgrade isn't performed directly on the production system and users are not switched until the upgrade has been thoroughly tested and deemed successful.

**About the author**

Bill has worked with Oracle databases since 1988, and has presented several times at Oracle Open World and its predecessor. He has worked with the SharePlex replication solution for Dell as well as a customer, helping hundreds of customers increase their system availability and deploy scalable infrastructures. His experience with Oracle includes database administration, architecture, data warehousing, Oracle E-Business Suite and application development. Including hands on experience with replication technology like change data capture, messaging queues, block level disk writes, enterprise application integration, extract transformation and load processing.

SharePlex customers using the migration process outlined in this paper have reduced their downtime by an average of 94 percent.



## For More Information

© 2014 Dell, Inc. ALL RIGHTS RESERVED. This document contains proprietary information protected by copyright. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose without the written permission of Dell, Inc. ("Dell").

Dell, Dell Software, the Dell Software logo and products—as identified in this document—are registered trademarks of Dell, Inc. in the U.S.A. and/or other countries. All other trademarks and registered trademarks are property of their respective owners.

The information in this document is provided in connection with Dell products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Dell products. EXCEPT AS SET FORTH IN DELL'S TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT,

DELL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DELL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF DELL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Dell makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Dell does not make any commitment to update the information contained in this document.

## About Dell Software

Dell Software helps customers unlock greater potential through the power of technology—delivering scalable, affordable and simple-to-use solutions that simplify IT and mitigate risk. The Dell Software portfolio addresses five key areas of customer needs: data center and cloud management, information management, mobile workforce management, security and data protection. This software, when combined with Dell hardware and services, drives unmatched efficiency and productivity to accelerate business results. [www.dellsoftware.com](http://www.dellsoftware.com).

If you have any questions regarding your potential use of this material, contact:

### Dell Software

5 Polaris Way  
Aliso Viejo, CA 92656  
[www.dellsoftware.com](http://www.dellsoftware.com)

Refer to our Web site for regional and international office information.