



Real-time Data Replication

from Oracle to other databases
using DataCurrents



WHITEPAPER



Contents

- Data Replication Concepts 2**
- Real-time Data Replication 3**
- Heterogeneous Data Replication..... 4**
- Different data replication technologies by Oracle 5**
 - Distributed Transactions 5
 - Materialized views (Snapshots) 5
 - Multi-Master Replication 5
 - Updateable Materialized Views 6
 - Oracle Advanced Replication 6
 - Oracle Data Guard..... 6
 - Oracle Warehouse Builder 6
 - Oracle Streams Data Replication 6
 - Oracle Heterogeneous Services and Transparent Gateways..... 7
 - Oracle Golden Gate..... 7
- DataCurrents as a convenient and affordable alternative..... 8**
- Appendix A 10**

Data Replication Concepts



There are many examples when business dictates requirements for copying data from one database to another. One of the most common examples is when a production OLTP database needs to be off-loaded with reporting activities and data copied to a separate database, or moved over to remote departments.

In this or similar cases, data is being copied from a source database to a target database periodically or continuously, synchronizing and maintaining consistency. This is called **data replication**, which means the creation (and maintaining) of data replicas.

There are many ways to perform required data copying using home-grown SQL scripts, the tools, technologies, and facilities that are available in source or target databases, or by using independent third-party products.

This whitepaper provides overview of data copying opportunities. Custom or “home-grown” SQL scripts are not covered. Although the scripts can be crafted and adjusted to meet business requirements, the development and maintenance for the scripts can be an expensive and error-prone task. Maintenance and manageability of such scripts is usually poor because of their complexity and lack of documentation.

DataCurrents is the cost effective and easy-to-use real-time data replication solution between Oracle to Oracle, Microsoft SQL Server, MySQL and other JDBC compliant databases. DataCurrents does not require any changes within source replicated tables and it supports all OS platforms and editions of source Oracle databases.



Real-time Data Replication

Real-time replication requires data to be copied over and distributed continuously, right after changes are applied to the data source. Real-time replication can be synchronous and asynchronous.

Synchronous (Real Time) – a data change is propagated to other database as a part of the same transaction; in this case each transaction becomes distributed. Data change becomes visible (committed) only after it is successfully applied to both databases. If a data change fails in the remote database, it will also fail in the primary database.

Asynchronous (Close to Real Time) - a data change is propagated to other database as a separate transaction. Data change becomes available in each database separately: first in the local database and then, after successful propagation, in another database. Propagation usually takes time, which creates data replication latency.

Note that for Asynchronous replication, data change will be still be available if a corresponding change is delayed or even fails in the target database. Local data changes must be stored before being propagated to other databases.

Both replication types have their strengths and weaknesses.

	Strengths	Weaknesses
Synchronous	<ul style="list-style-type: none"> Consistency is guaranteed Consistency is supported by internal distributed transaction mechanism No visible propagation latency 	<ul style="list-style-type: none"> Target database availability impacts source/primary database transactions May require manual involvement during distributed transactions (two-phase commit) failures. Replication into heterogeneous databases requires external distributed transaction broker It may significantly slow down transactions in the source database if a target database is remote
Asynchronous	<ul style="list-style-type: none"> Creates little impact on local database Does not slow down transactions/DMLs in the source database Source database is not dependent on availability of a target database 	<ul style="list-style-type: none"> Point-in-time consistency between primary and target databases is not guaranteed Data is synchronized with some latency Requires to store data changes from primary database

Heterogeneous Data Replication



In a generic case, the source and target databases are the same; homogenous. All major vendors usually have built-in replication technologies for their database systems. For example, database links in Oracle or Linked Servers in Microsoft SQL Server.

When the source and target databases are from different database vendors, the replication technology must be compatible with both databases. The replication technology across heterogeneous databases is often available as an optional add-on and is usually licensed separately (for example, Oracle Transparent Gateways).

DataCurrents is a single all-in-one solution for heterogeneous replication from Oracle databases to other Oracle as well as non-Oracle databases. It offers built-in support for Oracle, Microsoft SQL Server and MySQL as targets and custom configuration support for DB2, Sybase, and other JDBC compliant databases such as Teradata and Postgres.

Different data replication technologies by Oracle



Oracle provides multiple technologies, tools, and APIs to replicate data from one database to another. The most common technologies for Oracle data replication are:

- Distributed Transactions
- Materialized views (snapshots)
- Multi-master replication
- Updatable Materialized Views
- Oracle Advanced Replication
- Oracle DataGuard
- Oracle Warehouse Builder
- Oracle Streams
- Oracle Heterogeneous Services and Transparent Gateways
- Oracle GoldenGate

Below are various technological issues for various Oracle Replication options. These issues illustrate why inexperienced Oracle DBAs may be confused as to which tool to use for which replication task—which adds complexity, time, and unnecessary costs in daily database administration. In most cases, DataCurrents will remove this confusion.

Distributed Transactions

Administrators and users can create a link to a remote database and a trigger on a source table with required DMLs to the trigger definition that need to be executed in the remote database via the database link. Oracle will implicitly distribute such transactions and will use two-phase commit to ensure that both changes, the local and the remote, are either committed or rolled-back at the same time.

Materialized views (Snapshots)

Administrators and users can create materialized views to replicate data from remote databases via a database link; they can also be scheduled to periodically auto-refresh. A materialized View Log can be created on source table to register rows that have been changed. In this case, Materialized Views can refer to the Materialized View Log to replicate only changed records in order to minimize replication/refresh time.

Multi-Master Replication

Multi-Master Replication is a peer-to-peer replication between the databases (sites). Each site can push data changes to a one or multiple peer sites. Multi-Master Replication can be synchronous and asynchronous. With asynchronous replication data changes are stored locally on a site as deferred transactions and then propagated to peers. Oracle provides conflict resolution functionality when data for the same row is changed on both sides.

Updateable Materialized Views

Updateable Materialized Views allow modifying data in source Materialized Views. Data changes saved as a deferred transactions will be propagated back to the source table during a refresh. Updateable Materialized Views are a combination of a Materialized Views with Multi-Master Replication for back propagation of changes. Oracle provides conflict resolution functionality when data for the same row is changed on both sides.

Oracle Advanced Replication

Advanced Replication is an option that includes Multi-Master Replication, Materialized Views, Updateable Materialized Views, Replication Groups, Replication Sites, PL/SQL packages and GUI for management.

Oracle Data Guard

Oracle Data Guard is not exactly a data replication, but rather a high availability and disaster recovery solution for continuous copying of the entire production database into stand-by database. If the production database becomes unavailable because of a planned or unplanned outage, Data Guard can switch any standby database to the production role.

A standby database can be physical or logical.

The **physical standby** is a physically identical copy of the primary database. Data Guard continuously synchronizes the stand-by database by applying a redo-log of the primary database to it.

The physical standby is usually closed for users, and it can process read-only (SELECT) SQL statements using Active DataGuard mode, which is a separately licensed option for DataGuard.

The **logical standby** database has the same logical structure as the primary one, but physically it is a different database and can have different physical structures (for example, data files). Data Guard extracts SQL statements from the redo-log of the primary database and applies/executes them on the standby database.

Logical standby database can be used for query purposes (i.e. reporting).

Oracle Warehouse Builder

Oracle Warehouse Builder (OWB) is an ETL – data integration, data warehousing, data quality and metadata management solution. OWB is an integral part of Oracle Database 11g Release 2.

OWB includes a GUI designer where ETL packages can be created using graphical UI and then translated into PL/SQL packages. The PL/SQL packages do actual extraction and propagation of the data with possible transformation and quality management (aka cleansing).

Oracle Streams Data Replication

Oracle Streams Data Replication uses Oracle Streams framework to perform arbitrary data sharing via reliable message propagation. Oracle Streams, in its turn, is based on Oracle Advanced Queues – an

integrated feature of Oracle databases. Messages are queued (or stored into queues) during the propagation process.

Using Streams Data Replication, each data change is captured and wrapped into a Streams message that is propagated and applied to a target database.

As per Oracle documentation, Streams functionality will not be enhanced in future Oracle releases.

Oracle Heterogeneous Services and Transparent Gateways

Oracle Heterogeneous Services is a technology that can be used for data replication.

Oracle Heterogeneous Services (HS) is an integrated feature of Oracle databases. It allows connecting Oracle with non-Oracle databases via transparent gateways, or via generic (ODBC or OLE DB) agents. Heterogeneous Services translate Oracle SQL statements into the proper dialect of the non-Oracle system, as well as presents metadata of non-Oracle systems in the Oracle dictionary format. Oracle HS with transparent gateways also manage global transactions to ensure transactional integrity.

Oracle Golden Gate

Oracle GoldenGate is an asynchronous, log-based, real-time data replication product that moves transactional data in real-time across heterogeneous database, hardware, and operating systems.

Oracle recommends GoldenGate as a long-term replication solution.

DataCurrents as a convenient and affordable alternative



DataCurrents is a real-time, asynchronous data replication solution that propagates transactional data changes into heterogeneous databases.

DataCurrents can also be used to migrate large volumes of constantly changing data, helping to minimize application downtime. For example, large application databases can be copied over to a target database while the application is online and updating database tables. After data copy is completed, the application can be switched to the new database with a very short downtime.

DataCurrents is a multi-platform software product and has a flexible and powerful architecture, capturing data changes in real time with minimal impact on the source database. The data capturing functionality within DataCurrents relies on the native Oracle API that guarantees compatibility and supportability with future Oracle RDBMS versions.

DataCurrents can perform conversions of data from source types to the target types implicitly or by using custom expressions. It includes an intuitive GUI: the DataCurrents Administration Console, which allows adding or removing tables into or from the replication context, customizing target tables, and data conversions. Adding, removing or refreshing the tables can be done completely online without impacting running applications.

Data transfer and data replication functionality within DataCurrents preserves original transactional consistency.

DataCurrents reports its current status and statistics: latency, the number of DMLs captured, applied, and others.

Benefits of DataCurrents:

- Very close to real time with sub-second average latency
- Supports Oracle, MS SQL, My SQL and, other databases
- Virtually zero impact on the source DB; doesn't require any changes within source replicated tables; no triggers or additional columns
- No interruption or downtime for source and target databases associated with the software installation, deployment and administration
- Fully supports transactional integrity
- Implicitly converts all major data types
- Flexible—can use expressions for custom “in-flight” data conversions during data replication
- Supports bi-directional replication
- Includes intuitive and ergonomic GUI (Administration Console), which has convenient and effective wizards for adding tables into replication context
- Does not require intense training

- Very small deployment time—usually takes only minutes to install and set replication for hundreds and thousands of tables
- Flexible architecture: the DataCurrents service can be installed on source, target or a separate system; Administration Console (GUI) can be installed anywhere
- DataCurrents Administration Console wizards enable the following options:
 - Modification of target table name, columns, data types, expressions and storage options
 - Optional copy of primary, unique and foreign keys and indexes
 - Optional initial data loading
- Small CPU, memory and space footprint
- Does not require full refreshes for target tables; if needed, such refreshes can be done manually; they are transaction consistent even if the source table data changed during the refresh (source table is not locked during refresh)
- Supports parallelism with adjustable degree if manual refresh of tables is required
- Supports all OS platforms that support Java
- Supports all editions of source Oracle databases including free Oracle XE
- Supports all editions of Oracle, MS SQL, MySQL target databases including free ones such as Oracle XE and free Microsoft SQL Server Express
- Licensing, maintenance, and support is very cost effective

Appendix A

Below are different ways of data replication in Oracle.

Replication using	Is Replication Solution	Does not need coding	Supports Real Time	Supports bi-directional	Supports implicit data type conversion	Can create default tables	On-line adding/removing of tables	Has GUI console	Is heterogeneous
DataCurrents	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Materialized views (Snapshots)	No	No	No	No	Yes	Yes	N/A	No	No
Multi-Master Replication	Yes	No	No	Yes	Yes	No	N/A	No	No
Updateable Materialized Views	No	No	No	Yes	Yes	No	N/A	No	No
Oracle Advanced Replication	Yes	No	Yes	Yes	Yes	No	Yes	No	No
Oracle Data Guard	No	Yes	Yes	No	Yes	No	No	?	No
Oracle Warehouse Builder	No	No ¹	Yes	Yes	Yes	No	No	Yes	Yes ²
Oracle Streams Data Replication	Yes	Yes ³	Yes	Yes	Yes	No	Yes	No	Yes ²
Oracle Heterogeneous Services and Transparent Gateways	No	No	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Oracle Golden Gate	Yes	Yes ³	Yes	Yes	Yes	Yes?	No	Yes?	Yes

¹ – Coding is graphical. The ETL-like diagrams should be created using graphical environment, after that the diagrams will be translated to PL/SQL packages which should be deployed on source or target.

² – Working with non-Oracle databases will require either Heterogeneous Services or Transparent Gateway.

³ – When fails the restart, requires drop and re-create replication, PL/SQL API scripts can be used to avoid tedious GUI work