

# the *Availability Digest*

[www.availabilitydigest.com](http://www.availabilitydigest.com)  
[@availabilitydig](https://twitter.com/availabilitydig)

## Attunity Replicate

August 2014

Attunity Replicate is a data replication engine that supports homogeneous and heterogeneous database replication. It replaces Attunity Stream. Replicate supports most relational and non-relational databases as sources and many of these as targets.



Replicate provides initial target table creation using the source-database schema, full initial table load while the source applications are running, and real-time DML (Data Manipulation Language) and DDL (Data Definition Language) replication.

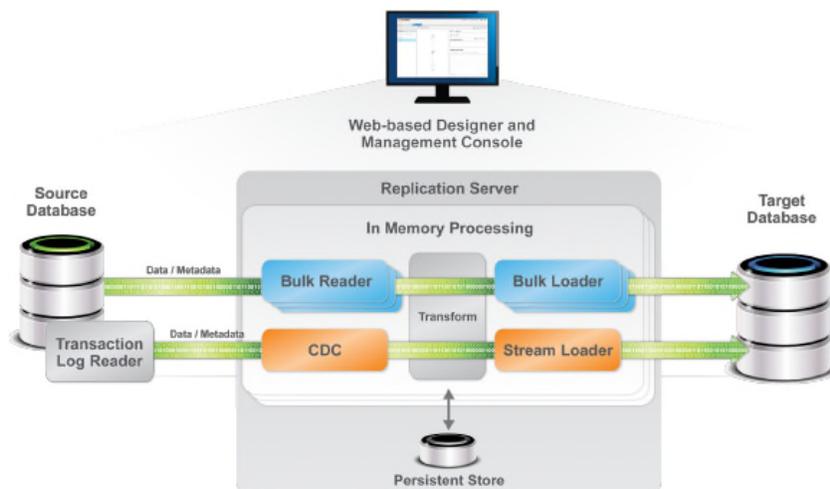
Replicate uses memory-resident streaming to achieve second and subsecond replication latency. It is multithreaded for high scalability. It provides web interfaces for configuration and management.

If the database provides an external interface (such as ODBC or JDBC) to either the source or the target database, no software agent need be installed on these systems. Rather, Replicate is installed on a Windows or Linux Replication Server. Attunity calls this configuration its “zero-footprint” feature. Alternatively, Replicate agents may be installed on the source and/or target systems. If Replicate agents are installed on both the source and target systems, the intermediate Replication Server system is not required.

### Replicate Architecture

Replicate is organized into three domains:

- the source database.
- the Replication Server (if required).
- the target database.



## **Replication Server**

In its “zero-footprint” configuration with no agents stored on either the source or target systems, Replicate uses a stand-alone Replication Server running Linux or Windows to read the change log of the source system and to apply changes to the target system. Changes for a transaction are buffered in the Replication Server and are applied to the target database only as completed transactions. Therefore, the target system always provides data integrity.

The Replication Server can perform data transformations on the data as it replicates it. Web-based interfaces are available to configure the replication (Click-2-Replicate Designer) and to manage the replication process in real time (Attunity Replicate Console).

Multiple multithreaded Replication Servers can be configured to achieve whatever throughput is required. A Replication Server can be configured in a bidirectional mode to support active/active architectures. To do this, two replication threads are created in the Replication Server, one for replication in each direction.

The Replication Server may run on any of the following platforms:

- Linux Red Hat
- Windows Server 7 x64
- Windows Server 2008 R2 x64
- Windows Server 2012 and 2012 R2 x64
- Windows 7 x64

The Replication Servers can be configured as a redundant pair in order to avoid a single point of failure in the replication channel.

If the source and target Replicate agents are installed on the source and target systems, the intermediate Replication Server system is not needed. Replication proceeds directly via communication from the Replicate agent on the source system to the replication agent on the target system.

## **Bulk Loading**

The first step in establishing replication with Replicate is to create copies of the source database structures on the target system. The target database metadata is derived from the source database metadata, with appropriate transformations defined by the user. The source data is then read from the source-database tables and files via the Replicate Bulk Reader, and these tables and files are copied into the target database via the Bulk Loader.

There is no need to stop the source applications during the load process. Rather, changes to the source database during the load are buffered and are applied to the target database after the initial load has completed. Therefore, the target database may be inconsistent during the load but will be fully consistent following the application of all changes that occurred during the load.

If changes cannot be applied to the target database in a reasonable time, they are buffered until they can be applied. This avoids the roll-over problem in which the change log is archived and is no longer available online to provide changes.

The bulk load can be interrupted at any time and then restarted. New tables can be added during the load, and columns can be added or dropped without the need to reload.

## **Change Replication**

Replicate replicates both data (DML) changes and schema (DDL) changes in real time. All changes can be subject to transformation as replication proceeds.

## Data Changes

Replicate obtains a stream of data changes (DML changes – inserts, updates, deletes) from the source system's change logs. Replicate buffers the changes within the scope of a transaction until it has received the complete transaction and then applies the transaction to the target system. In this way, the target system always maintains data integrity.

The source system's change logs are read by the Transaction Log Reader, which sends the changes to the Change Data Capture (CDC) facility resident in the Replication Server. If the source system provides a means for the Replication Server to read the change log, no Replicate agent need be present on the source system. The Replication Server can read the changes directly. Otherwise, the Transaction Log Reader is installed on the source system as a Replicate agent to read the change log and to feed the changes to the CDC resident on the Replication Server.

When a complete transaction has been assembled in the Replication Server, it is loaded into the target database via the Replicate Stream Loader. If the target database provides external access (for instance, via ODBC or JDBC), the Replication Server updates the target database directly with no need for a Replicate agent on the target system. Otherwise, a Replicate agent is installed on the target system to apply replicated transactions to the target database.

The location of the last replicated transaction is maintained. This allows Replicate to recover rapidly from interruptions and failures and to guarantee transaction delivery.

Replicate replicates structured, semi-structured, and unstructured databases. If the source system uses a non-relational database that does not create a change log, a Replicate agent is installed to capture changes to the database and to send them to the Replication Server.

Replicate provides homogeneous and heterogeneous replication. Data can be replicated from any supported source database to any supported target database. The stream of events read from the source change logs can be filtered and transformed to meet the needs of the target database.

## Schema Changes

Replicate also replicates source database schema (DDL) changes to the target database. This includes adding or deleting files or tables, adding or deleting columns, and making changes to data formats.

Schema replication is used initially to create the target database structures prior to a bulk load. Once the target database has been loaded and is operational, schema replication continues to keep the target database structure synchronized with that of the source database.

## Transformation

Replicate uses the source database metadata and the target database metadata to map source database changes into appropriate target database changes. A subset of the tables and columns can be selected for replication. Content-based row-level filtering can be defined to avoid replicating unimportant data.

According to user specifications entered via the Click-2-Replicate facility, data can be transformed between the source and target systems. Tables and columns can be added, deleted, or renamed. The data type or length of any column can be changed.

For transformations not supported by Replicate, there are currently no means provided for the user to insert code for specialized transformations into the Replication Server's transformation facility. However, this feature is on Replicate's roadmap.

## Zero-Footprint

As described above, if the source and target systems offer means for external access of their databases, no Replicate agent need be installed on them. Rather, the replication functions (source and/or target) are provided by the Replication Server system. Attunity calls this its “zero-footprint” architecture.

The source system requires an interface that allows CDC to read the change logs directly. If no interface is available, or if the vendor-supplied interface is insufficient or inefficient, Replicate installs an agent on the source system to capture changes and to send them to the Replication Server. This agent may be the Transaction Log Reader.

Likewise, if the target system is not provided with an interface for directly updating the target files or tables, or if the vendor-supplied interface is insufficient or inefficient, Replicate will install an agent on the target system to apply transactions to the target database. If a Replicate agent is installed on both the source and target systems, the Replication Server system may not be required.

Examples of database access facilities that can be used by Replicate to directly access a database include JDBC (Java Database Connectivity), ODBC (Open Database Connectivity), and Microsoft’s .Net.

## **Scalability**

Replicate is designed to be highly scalable. Change replication is via in-memory streaming. There are no intermediate disk-storage points, so replication latency (the time from when a transaction is committed to the source system to the time that it is applied to the target database) is minimal. Attunity claims replication-latency intervals in the second to subsecond range.

Multiple replication threads can be deployed on the same Replication Server hardware platform to provide additional replication capacity. Each replication thread is assigned specific tables or files to replicate. All related tables or files must be replicated by a single replication thread since transactions flowing over separate replication threads may be completed out of order at the target database. For instance, if a parent table is replicated by a different replication thread than its child table, transactions adding a child may occur before the parent is created, resulting in a violation of referential integrity.

Multithreading can also be used to implement redundancy of the Replication Server hardware platforms. Two Replication Servers can be provided – one replicating one set of tables and the other replicating the other tables. Should a Replication Server fail, the surviving Replication Server takes over the replication of all tables.

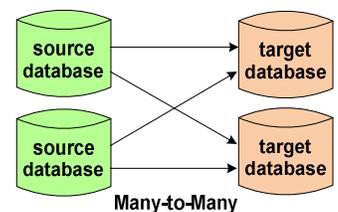
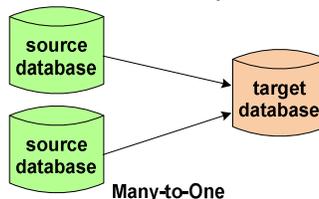
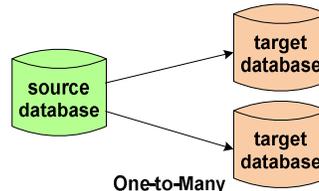
The use of multiple threads in the same Replication Server or in multiple Replication Servers also provides fast parallel loading for data warehouses.

Replicate optimizes the communication of changes over WAN networks. Attunity’s TurboStream DX is a WAN transfer engine that optimizes transfer speeds to target databases located off-premise or in a cloud.

## **Topologies**

Replicate supports several topologies:

One-to-One: A single source database is replicated to a single target database.



One-to-Many: A single source database is replicated to multiple target databases.

Many-to-One: Multiple source databases are replicated to a single target database.

Many-to-Many: Multiple source databases are replicated to multiple target databases.

Bidirectional: Changes are replicated in both directions between two databases.

### ***Active/Active Architectures***

Active/active architectures are supported by Replicate's bidirectional replication. To implement bidirectional replication, two replication threads are configured on the same Replication Server. Each replication thread transfers changes from one database to the other but in opposite directions. Thus, a change made to either database is replicated to the other database; and both databases reflect the same application state. Therefore, a transaction can be sent to either node and be processed in the same way.

The transaction load is shared between the nodes. Should one node fail, all that needs to be done is to reroute all transactions to the surviving node. Recovery from an outage can be accomplished in seconds, thereby ensuring true continuous availability.

The detection and resolution of data collisions (both nodes updating the same data object at the same time) is not currently supported. However, this function is on the Replicate roadmap.

### ***Web Interfaces***

Attunity Replicate is configured and managed by web interfaces. These functions can be performed via a web browser from any location.

The Click-2-Replicate Designer is used to configure replication. The Attunity Replicate Console monitors and manages a replication configuration.

#### Click-2-Replicate

The Click-2-Replicate web service allows the user to define the structure of the replication channel via a simple drag-and-drop GUI.

The database schemas for the source and target databases are used to configure replication. The columns to be replicated are selected, and the transformations required between columns are specified. Row content filtering can be defined.

When the definition of the replication channel is complete, the resulting Replicate configuration can be deployed.

#### Attunity Replicate Console

The Attunity Replicate Console web service is used to monitor and manage Replicate instances in real time from a web browser.

Tasks can be started, stopped, and reset at any time from any location. As a monitoring service, the Replicate Console acts as a real-time dashboard. It displays such information as the change throughput, the replication latency, and the numbers of replicated inserts, updates, deletes, and metadata changes.

The Replicate Console also displays alerts about the status of the replication channel and the various replication tasks.

## **Activity Log**

Replicate provides an activity log of all actions taken to configure and manage the replication channel as well as all alerts generated concerning the status of the replication channel. The log supports the integration of Replicate with enterprise-management systems.

## **Supported Databases**

Replicate supports replication between several relational and non-relational databases as well as replication between on-premise systems and clouds and replication to data warehouses.

### Databases

The following databases are supported:

- Oracle is supported as a source database and as a target database. Changes are captured from the Redo Log using the Oracle BFILE log access facility.
- SQL Server is supported as a source database and as a target database. Changes are captured from the TLOG (Transaction Log).
- MySQL is supported as a source database and as a target database.
- SAP Sybase ASE (Adaptive Server Enterprise) is supported as a source database and as a target database.
- HP NonStop SQL/MP is supported as a source database and as a target database (SQL/MX is not supported). Changes are captured from the Audit Trail. Replicate agents are required on both the source and target systems.
- DB2 is supported as a source database.
- Salesforce is supported as a source database.
- Any database with an ODBC or JDBC interface to the database is supported as a target database.
- The following legacy databases are supported using Attunity's Integration Suite agents:
  - VSAM
  - IMS/DB
  - HP NonStop Enscribe
  - OpenVMS RMS

### Clouds

Replicate supports data replication to and from the Microsoft Azure Cloud and the Amazon AWS (Amazon Web Services) Cloud.

### Data Warehouses

The following data warehouses are supported:

Action Vectorwise  
Amazon Redshift  
HP Vertica

IBM Netezza  
Microsoft PDW  
Oracle Exadata  
Pivotal Greenplum  
SAP Sybase IQ  
Teradata

## Use Cases

Attunity promotes Replicate for many uses, including the following:

Offloading critical, processing-intensive functions to other systems:

- business intelligence
- analytics
- operational reporting
- query processing

Feeding data warehouses and data marts with real-time information.

Generation of change tables for ETL (Extract, Transform, Load) utilities.

Achieving high- and continuous availability:

- synchronizing a disaster-recovery site with the production site.
- zero-downtime migrations and upgrades.
- active/active systems for continuous availability.

Consolidating data from several systems onto a common system.

Distributing data to provide proximity to geographically dispersed systems.

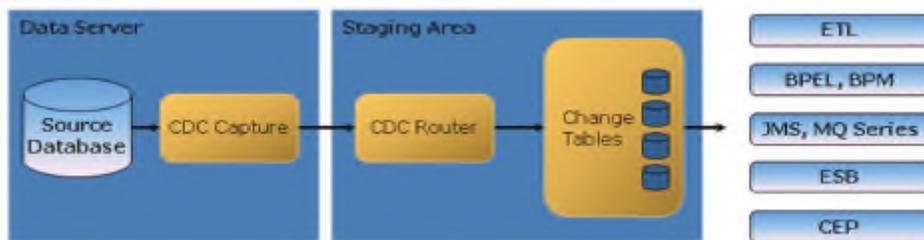
Segmenting data for security purposes.

Generating test data for test systems.

Migrating data between on-premise systems and clouds for cloud computing.

## Change Data Capture

Attunity's Change Data Capture facility is central to its Replicate data replication engine. However, CDC is also used to send data to a variety of operational data stores via various standard interfaces.



A CDC Capture agent runs on the source system. It reads changes from the source system's change log and identifies those changes that must be forwarded. It sends these changes to a staging area running on a separate Windows or Linux system.

In the staging area, a Router applies intelligent filters and stores the resulting changes in Change Tables. From there, the changes are available for consumption by outside services. These external services include Extract, Transform, and Load (ETL) utilities, Enterprise Application Integration (EAI) utilities, Business Process Management (BPM) utilities, Business Process Execution Language (BPEL) utilities, Complex Event Processing (CEP) utilities, and homegrown applications driven by JMS (Java Messaging Services), MQ Series, or an Enterprise Service Bus (ESB).

Changes are made available to these services either through polling or publication. With polling, a service accesses changes via SQL queries. With publication, a service subscribes to the change data and is sent the data as it becomes available via XML messaging.

CDC provides all metadata management for the data transfers. Data models for the data being captured are imported. If the data source is non-relational, the source metadata is mapped to a relational model for distribution.

CDC supports mainframe, Windows, Linux, HP NonStop, OpenVMS, and AS/400 platforms.

## Summary

Replicate is a powerful data replication engine that can be used to synchronize homogeneous and heterogeneous databases. The databases may be relational or non-relational. Powerful transformation facilities support the conversion of source database formats to those of the target database. All DML and DDL changes can be replicated with replication latency measured in seconds or subseconds.

The initial target database can be created and loaded without having to pause the source applications. Web-service GUI consoles are provided to initially define and deploy the replication channel and then to monitor and manage it.

Replicate Replication Server appliances can be multithreaded to provide scalability. Multiple Replication Servers can be provided to meet any capacity requirements and to provide redundancy to avoid single-points-of-failure in the replication channel.

Several topologies are supported, including active/active systems for continuous availability. Replicate supports the integration of on-premise systems to cloud services.

Replicate is currently deployed to offload production systems by moving process-intensive functions to other systems, to provide disaster recovery to remote data centers, and to feed data warehouses, data marts, and Extract, Transform, and Load utilities, among many other uses.