

# Comparing Microsoft SQL Server 2005 Replication and DataXtend Remote Edition for Mobile and Distributed Applications

White Paper



[www.dxstro.com](http://www.dxstro.com)

---

## Table of Contents

Overview .....	3
Replication Types Supported.....	3
Set-up & Configuration .....	4
Replication Network Topology Options.....	4
Maintaining Ongoing Synchronization .....	4
Capturing Changes in the Database .....	5
Managing Changes to the Replication Network .....	5
Selecting Data for Replication .....	6
Managing Globally Unique Identifiers (GUIDs).....	6
Supporting Multiple Database Types.....	6
Support for Complex, High-Volume Database Applications.....	7
Specific Capabilities Not Found in Microsoft SQL Server 2005.....	7
File Replication .....	7
Replication Horizon and Metadata Management .....	7
Exception Handling.....	7
Site Pre-Configuration .....	8
Advanced Scheduling .....	8
Conclusion .....	8
About DXSTRO .....	9

---

## Overview

DataXtend™ Remote Edition and Microsoft® SQL Server™ both provide a set of replication tools for copying, distributing, and synchronizing data from one database to another. DataXtend RE (Remote Edition) provides the most comprehensive and flexible replication and synchronization tools available for software developers and enterprise IT departments to make their applications mobile and available both on-line and off-line. While SQL Server replication technology can be used for replicating data for reporting, backup and recovery, or mobile applications, its features are better suited for non-mobile, non-distributed/connected applications.

If you are considering Microsoft replication for a mobile or distributed application, this whitepaper will help you understand which features may and may not be appropriate before investing valuable development resources.

## Replication Types Supported

There are three types of replication available with Microsoft SQL Server. These are snapshot replication, transactional replication and merge replication. Each replication type provides different capabilities depending on the application, and different levels of ACID properties of transactions and site autonomy. Snapshot replication is the process of copying and distributing data exactly as it appears at a moment in time. This may be helpful in situations where data is rather static or infrequently changes. With transactional replication, an initial snapshot of data is propagated to Subscribers, and then when data modifications are made at the Publisher, the individual transactions are captured and propagated to Subscribers. This would be useful in applications where changes must be replicated quickly to subscribers, but may require a nearly constant network connection for installations where a high rate of change persists and/or limited disk resources exist.<sup>1</sup> With merge replication, an initial snapshot is applied to the distributed sites; subsequent changes to the database are tracked, then published to all subscribers. This allows the replicated sites to work online or offline and merge data modifications made at multiple sites into a single, uniform result at a later time.

Merge replication synchronizes pairs of sites to the same data values. As in any distributed environment, as other sites may be simultaneously changing, ACID properties cannot be assured. Data values in the replication topology converge to the same value over time.

These characteristics make merge replication the better solution for distributed and mobile applications where users need full read/write access to local replicas of data. The remainder of this whitepaper will primarily evaluate Microsoft's merge replication features.

---

## Set-up & Configuration

Microsoft merge replication is implemented by the Snapshot Agent and Merge Agent. The Snapshot Agent prepares snapshot files containing schema and data of published tables, stores the files in the snapshot folder, and inserts synchronization jobs in the publication database. The Snapshot Agent also creates replication-specific stored procedures, triggers, and system tables.

While many of the setup steps are automated, some of the most complex steps are not. For example, if the snapshot files are going to become so large as to require an FTP-based file transfer, rather than UNC paths, manual setup of this transfer mechanism is required. This would require numerous, manual configuration steps on each of the involved hosts. Many services used in Microsoft Replication depend upon unified authentication rules across the enterprise. Without deep knowledge of the networking environment, one might require having a Microsoft network administrator close by. Particularly when implementing Microsoft Replication for distributed organizations, it is likely that there are disparate LAN environments or services running as different users.

Initial installation of DataXtend RE replication is a simple point-and-click process. Optionally, for embedded deployment, it can be installed via a single command line. Once installed, application databases can be enabled for DataXtend RE replication via a single command line. There is no requirement for domain wide privileged accounts. Each site's replication service uses its own local account with local Administrator privileges. Operation across domains or even without a domain is not cause for concern.

## Replication Network Topology Options

The replication model, or topology, is the logical map for how data will be distributed across the enterprise and for how the servers will be configured during replication implementation. Microsoft primarily supports a "hub & spoke" network with a central Publisher and multiple Subscribers. Replicated data flows from the Publisher, possibly through a Distributor, and to the Subscribers. Similarly, updates flow back through the network to the Publisher.

DataXtend RE supports multiple replication network configurations and topologies, including Peer-to-Peer, Hub & Spoke, and Master-Slave. Additionally, even more complex replication topologies are possible in order to meet the business and service level requirements by combining or nesting the aforementioned topologies. For example, you could combine the Peer-to-Peer model with Master-Slave to create a "Cloud & Spoke" topology. The "cloud" is essentially a group of peer-to-peer nodes acting as a logical hub. This provides additional benefits such as load balancing, failover, and inherent back-up databases at the hub level.

## Maintaining Ongoing Synchronization

The Microsoft merge replication architecture is based upon an initial and periodic snapshot-and-restore process to re-synchronize the sites in the network. Although the snapshot creation and application process is largely automated, the data transfer is made using Microsoft Networking protocol. Microsoft recommends that the snapshot files be transferred via a more efficient protocol, such as FTP. These snapshot files can be very large and require adequate storage. For the average, realistically sized application database, this process can be very time consuming and represents unnecessary 'down-time'. It may also be physically impossible to 'push out' new copies of a database to hundreds of users – particularly users who may be disconnected.

Microsoft provides an option to compress snapshot files; however the default transfer protocol is not encrypted. Using Microsoft's recommendation for an alternative transfer protocol, one would need to set up separate encrypted FTP servers on each site to be able

---

to achieve integrated, encrypted, compressed data transfer. Further the additional FTP server infrastructure adds complexity to a replication network.

Microsoft's recommendation to reload and re-synchronize the databases periodically will be extremely problematic for many, if not all, distributed applications. In addition, Microsoft recommends running validation routines to ensure that "data at the subscriber matches data at the Publisher". Even though this extra processing can run concurrently with normal database operation; particularly in the case of Microsoft checksum validation, it is very resource intensive for the server so it must be scheduled for off-peak periods – which certainly may not be convenient for all users simultaneously in distributed organizations and may cause loss of worker productivity due to system downtime.

DataXtend RE, on the other hand, continually exchanges updates between all replication partners using a compressed data stream which is secured using Triple DES encryption. These communications are made even more efficient by exchanging only the columns or sub-sets of columns which have changed since the last synchronization. The communications are also mindful of bandwidth availability and will send packets sized appropriately to minimize 'retries' and prevent data loss.

With DataXtend RE, it is not necessary to execute periodic database snapshots because a 'live' query, requesting what has changed, is performed at each site for every replication cycle. As a result, there is never a doubt that all data changes are detected and replicated. DataXtend RE's change capture mechanism ensures that no data will be lost without incurring the cost of database refreshes, while ensuring that each captured change is delivered and acknowledged by each site on the network.

## Capturing Changes in the Database

A replication solution's change capture mechanism is essential for monitoring changes and triggering synchronization. Microsoft's approach to change capture is to force the addition of new table columns within the application's database. For merge replication, new columns will be inserted into existing tables in the database, forcing an automatic, and non-configurable, modification to the underlying database schema. This obviously complicates future upgrades to the application and its database.

DataXtend RE's approach to capturing changes in the database is fundamentally different. Rather than forcing changes to the application's database, DataXtend RE implements a "shadow table" approach that stores timestamp and identifying information about those ongoing changes. By leaving the application data structure unmodified, application upgrades will be simpler, as there is no need to be concerned about how they will affect the replication architecture.

## Managing Changes to the Replication Network

With Microsoft replication, as publishing servers go offline or come online or when non-anonymous subscribers go offline or come online, other servers in the replication network may need to be reconfigured.

In contrast, DataXtend RE's approach is transparent with regard to replication partner selection. Leveraging a core peer-to-peer replication engine, nodes can instead enter and leave the network without reconfiguration and without the loss of data. As a result, planned or unplanned service windows become easier to manage. Once a site has been initially registered on a given replication network, no further administration is required in order to register its presence. When a site is on-line and available to participate in replication, DataXtend RE's partner selection algorithms automatically consider all available sites to make sure that all pertinent updates have been exchanged. Off-line sites will not be considered and any pending changes will be held until they are once again available.

---

## Selecting Data for Replication

Microsoft Replication is capable of distributing and synchronizing entire databases or sub-sets of databases. It allows vertical partitioning by defining the unit of replication as either the entire row, individual columns, or related rows across several tables. Microsoft replication does not however, support defining logical groupings of columns within a single table for the purposes of conflict detection and resolution. In some cases, "silent data corruption" is possible, by allowing conflicting updates to related columns to be made by independent sites.

Consider this example: although the fields "City", "State", and "ZIP code" may be individual columns in a particular table, they should be logically treated as a group. A change to State should coincide with a change to ZIP code and City. Changing State without also updating ZIP code and City would "corrupt" the integrity of the corresponding record. Microsoft does not provide native functionality to ensure that logical groupings of columns within a table remain intact in a conflict scenario. Instead, one would need to develop TSQL scripts and/or C++ code to accommodate this, being both code intensive and further complicating ongoing maintenance.

DataXtend RE replication, on the other hand, provides out of the box functionality to partition databases into logical groupings or data fragments. Using the DataXtend RE replication designer, multiple database columns or rows can be grouped and managed together to ensure consistent replication. Selecting data partitions for replication with DataXtend RE is managed with point-and-click operations in the replication designer tool. Given the common requirement for distributing and synchronizing particular "slices" of data in distributed environments, the graphical utility is designed to enable configuring complex structures, including many-to-many, subsets, and nested filtering.

## Managing Globally Unique Identifiers (GUIDs)

Microsoft replication automatically inserts unique identifiers into the database in order for replication to track each individual record. Even if the database already contains a unique identification number, Microsoft will force a change to the application - a change that otherwise would not be needed.

DataXtend RE's approach to replication is to provide a much less intrusive, more scalable mechanism for managing globally unique identifiers. Unlike Microsoft, DataXtend RE will not force an application schema modification. In addition, if the application already uses columns with the Identity property for this purpose, DataXtend RE can use them as-is and integrate them into the internal GUID mechanism to ensure a unique record identifier - regardless of the number of current or future sites.

## Supporting Multiple Database Types

A common requirement in distributed database deployments is the need for heterogeneous database replication, or the ability to seamlessly and bi-directionally replicate data between multiple database types. Not surprisingly, Microsoft replication only supports full-featured replication between Microsoft databases. Although limited heterogeneous replication capabilities are described, Microsoft can only support one-way replication to a non-Microsoft database at the subscriber. One way replication from an Oracle publisher to a SQL Server subscriber or distributor (transactional and snapshot replication only) is also supported.

In contrast, DataXtend RE permits each replication node to contain any database type in its native form. For example, a three-site replication network which needs to synchronize data between MS SQL Server, Oracle, and DB2 will need to only have one database instance of the appropriate type at each endpoint. Since only the native instance is necessary at each endpoint, the application can continue to interface with its data in the traditional fashion, just as if it were a centralized application with only one database instance - no special procedures or APIs are needed. This flexibility provides "database independence" and avoids locking the application into Microsoft SQL Server or its replication technology.

---

## Support for Complex, High-Volume Database Applications

Tables used in Microsoft merge replication can have a maximum of 246 columns and a maximum row size of 8060 bytes. If the database to be replicated exceeds, or may near, this size constraint, one should carefully reconsider using the native replication features.

With no database size restrictions, the DataXtend RE provides an alternative for complex, high-volume replication implementations.

## Specific Capabilities Not Found in Microsoft SQL Server 2005

### File Replication

DataXtend RE supports the replication of flat files outside the database. Microsoft does not support file replication. DataXtend RE imposes no restrictions on the number, type or size of replicated files.

### Replication Horizon and Metadata Management

DataXtend RE automatically identifies a replication horizon, a dynamic point in time at which all sites have received all the data subscribed to from all other sites. Change capture metadata older than the horizon can be deleted safely from all sites in the replication network. This keeps metadata size to a minimum, and provides useful information about replication frequency and the state of individual sites and the network as a whole.

Microsoft allows replication metadata to be purged after a user-configurable number of days, but provides no integrated mechanism to determine when it would be safe to delete such data. If change capture metadata is purged too soon, these changes will never be replicated, and may be permanently lost when the next snapshot is implemented.

### Exception Handling

DataXtend RE is often able to work around data server exceptions. For example, if a constraint violation occurs during replication DataXtend RE will note the exception, and will continue to replicate changes. Later in the same session the operation that generated the exception will be retried, often with positive results. DataXtend RE will continue to retry failed operations until no further progress is made.

This helps resolve cases where referential integrity is breached temporarily due to sorting or change ordering issues. For example, a child record may have been replicated before a parent. Or in the case where a record is deleted and shortly thereafter reinserted, a re-insert of the same record was attempted before the delete was processed. There are often numerous cases where retrying exceptions yields positive results and reduces administration.

Microsoft does not attempt to cycle back and retry failed operations during Merge Replication. This can dramatically increase the cost of supporting replication networks.

---

## Site Pre-Configuration

DataXtend RE's Pre-Configuration of Sites feature reduces the amount of time taken to deploy sites with similar attributes. This is accomplished by enabling administrators to pre-define replication schedules, work set subscriptions, database file relationships, and activation parameters. Administrators can pre-define these settings for single or multiple sites using site templates.

Each subscriber in a Microsoft Merge Replication network must be individually configured. Microsoft does not support the use of database templates as a shortcut for initializing subscriber sites sharing similar configuration profiles. This can increase deployment time when using Microsoft Merge Replication to a point where it is unsuitable for many environments.

## Advanced Scheduling

DataXtend RE includes advanced scheduling options, such as event driven schedules, which begin a replication session only when the local site or a specified partner site has data changes to replicate. This reduces empty cycle replication sessions and ensures timely delivery of new changes. Traditional "time" and "interval"-based schedule types are also supported.

Microsoft replication only supports traditional time-based scheduling where a publisher or a subscriber replicates at a given time or after a given interval.

## Conclusion

Microsoft SQL Server 2005 replication remains best suited for constantly connected, simple file synchronization. It does not have the capabilities required to fully support diverse mobile, distributed, and disconnected applications.

DataXtend RE continues to build upon industry-leading replication capabilities for diverse deployment environments. DataXtend RE also provides significant advantages over SQL Server 2005 in the areas of stability, management, and performance. These advantages play an increasing role as you deploy more sites and users and encounter low-bandwidth conditions.

---

## About DXSTRO

DXSTRO acquired the DataXtend RE software from Progress Software Corporation in early 2014.

The company is committed to developing the product further and supporting the many existing customers.

For further information go to [www.dxstro.com](http://www.dxstro.com) or telephone +44(0)845 408 4250



[www.dxstro.com](http://www.dxstro.com)