



*Checklist for Evaluators of Data  
Replication Solutions  
White Paper*

**PROGRESS**  
SOFTWARE

*Real Time Division*

## *Table of Contents*

Table of Contents.....	2
Abstract .....	3
Open System – Heterogeneous, Standards-Based .....	3
Scalability and Flexibility .....	4
Performance and Efficiency.....	5
Integrity and Reliability.....	6
Security.....	7
Ease of Use – for Developers.....	7
Ease of Use – for Runtime Administrators .....	8
Ease of Use – for End Users .....	9
About Progress Real Time Division .....	11



## *Abstract*

This document is intended as a guide for organizations who have committed to developing a distributed application environment and are evaluating various database replication solutions. It poses many of the common questions and supplies the appropriate answers from the Progress® DataXtend™ RE product perspective.

As a solution evaluator, we encourage you to add these questions to your own list and apply them to other solutions you may be considering.

## *Open System – Heterogeneous, Standards-Based*

⇒ How many databases does it support? Which ones?

Today, DataXtend RE supports Oracle, Microsoft SQL Server, Informix, Sybase ASE, Sybase SQL Anywhere, Progress, DB2 and PostgreSQL databases. Because DataXtend RE accesses the database as a client, it can replicate the high-end databases whether they are running on Windows, Unix, or other platforms. Some other products support only one database, or very few databases.

⇒ Can it replicate heterogeneously with different physical databases at different sites, without limitations?

DataXtend RE is a truly heterogeneous peer-to-peer environment and does not distinguish between “sources” and “targets.” It replicates bi-directionally between all supported databases with full “update-everywhere” capabilities and you can use different physical databases at different sites in the same distributed system.

Some other products claim to be heterogeneous, but really only support one or a few database(s) fully and the others with severe limitations (e.g., through gateways, non-bidirectional, and/or “target-only”).

⇒ Is it an Internet standard?

DataXtend RE is an official Internet standard service on TCP/IP ports 242 and 2584.

Replication products that are nonstandard are more difficult to use (or cannot be used) through firewalls.

⇒ Does it use standard security?

DataXtend RE uses industry-standard, government-endorsed Triple-DES, SHA, and ECC security algorithms through Certicom’s efficient Security Builder library.

Beware of products which use “custom” security or claim to have added security recently (see “*Security*” section).

⇒ Is it written using standard portable programming tools?

DataXtend RE was designed from the start for ease of portability to run natively on Linux, Unix and other platforms. The replication engine is written entirely in standard

C++. DataXtend RE uses Microsoft, Certicom and DataDirect libraries and tools supported on all popular computing platforms.

Replication products that are developed using non-portable languages or tools are difficult to port to other platforms in response to customer needs.

## *Scalability and Flexibility*

⇒ Does it support BLOBs?

DataXtend RE supports BLOBs, from small to multi-megabyte binary data.

Many replication products do not support BLOBs.

⇒ What type of replication does it use?

DataXtend RE supports true asynchronous update-everywhere replication. Sites connect in pairs, and each replication session bi-directionally transfers new updates to and from both sites.

Other products use message-based or synchronous replication. Both are weaker, even though they are sometimes touted as “features”. Message-based replication is weaker because: a) it is less efficient; b) it often relies on another transport like e-mail and ftp that does not always provide delivery time guarantees; and c) it is less robust, for example, sites cannot correctly determine and adjust for clock drifts. A replication product is incomplete if it relies on other transports (e-mail, ftp) in order to operate. Synchronous replication (the same or similar to traditional “distributed databases”) is weaker because it usually requires that all affected sites be available at the same time.

⇒ Is it truly peer-to-peer, or at least fully bidirectional? Or does it distinguish between “sources” and “targets”?

DataXtend RE is a true peer-to-peer system. Any site can replicate with any other that it is connected to over any TCP/IP network, including the Internet. Replication is symmetric and bidirectional, even if the two sites are running different databases (e.g., Progress and Oracle), or one site has much more data than the other (e.g., 50MB on a laptop and 50GB on a regional server). Sites simply exchange updates to whatever data they have in common. DataXtend RE does not enforce a distinction between “sources” and “targets,” but it does fully support non-peer configurations (see next question).

Most replication products distinguish between “sources” and “targets,” or “masters” and “slaves/replicas.” This can severely limit end-user usability and scalability, especially if it requires a centralized hub-and-spoke configuration where all databases must replicate with one central site. Also, many products support certain databases only if they are used as targets (“target-only”).

⇒ Does it allow “source/target,” hub-and-spoke, and/or unidirectional replication?

DataXtend RE supports all of these configurations, because they are just subsets of true peer-to-peer operation. For example, an information vendor may replicate specific subsets of a legal information database to specific customers who pay for that specific data, but the data may only ever be updated at one central site and “published” or “pushed” to the others.

Most replication products support these configurations, but are limited to these configurations only (see previous question).

⇒ Is it scalable to high-volume sites? Is it multithreaded and SMP-ready?

DataXtend RE is massively multithreaded and uses a separate thread for each replication session. It can run on a small notebook computer right on up to multiprocessor servers, and will immediately take advantage of the extra power. (Because of DataXtend RE’s efficiency, extra CPU power is almost never needed; this is most appropriate when the local database has frequent and very large changes, such as a terabyte-sized video

jukebox database whose videos are frequently updated. See the “Performance and Efficiency” section.)

⇒ Is it scalable to large numbers of sites?

Any DataXtend RE database is as easy to deploy to two sites as to two thousand. When a new site is installed, it performs an initial replication with any existing site, and then is a fully participating member of the database network with housekeeping information about the location, schedules, subscriptions, and other information about other sites. As machines are added or removed, DataXtend RE’s automatic adaptive load balancing transparently reconfigures the network dynamically for the most efficient propagation. No manual housekeeping is needed.

Most other products require an administrator to define which sites should replicate with which others in which order, SQL queries to define the subsets of data to reside at each site, and so on. This does not scale to networks of thousands of sites. (See the “Runtime Administrator Ease-of-Use” section.)

## *Performance and Efficiency*

⇒ Can it run well on machines with slower CPUs?

For a 5M row database which undergoes 10% change each day, DataXtend RE averages 8% CPU usage on a 700MHz Pentium III over the course of the day (and less, if replicating over a slower network so that replication is network-bound). When not replicating, DataXtend RE consumes negligible CPU time.

Other products that can visibly slow down the system while running are inappropriate for unattended transparent background operation for most users.

⇒ Does it replicate efficiently? Even over slow networks like WAN, Internet and cellular connections?

DataXtend RE replicates: a) only the changes (“deltas”) b) made since the last replication c) to the subset of the data that both sites have in common d) with full compression. With DataXtend RE’s “fragments” feature, this usually means replicating only small parts of a record, not full records, and allows true “trickle” and near-real-time replication. Sub-minute replications are common. For a 5M row local database which changes about 10% per day (~500k rows/day), replication typically requires less than 80 minutes per day at Internet or cellular-modem speeds.

Many products replicate only full “snapshots” of the database, complete images of all or much of the database’s data, even if little has actually changed. Others say that they replicate only changes, but actually transmit far more. Others use inefficient protocols that do not operate well over slow networks.

⇒ Does it allow concurrent replications of the same database?

DataXtend RE can replicate with arbitrarily many sites simultaneously. Concurrent sessions ‘follow’ each other through the schema, gathering and applying updates. This allows highly efficient ‘passthrough’ replication: For example, if site A is replicating with site B, and while that’s in progress B starts to replicate with site C, and C with site D, then the replication sessions overlap and updates travel from A to D nearly as quickly as if A and D were replicating directly.

Some other products are limited to a single replication at a time, which reduces scalability and usability. Even products which allow concurrent replications rarely support efficient ‘passthrough’ replication. (For more about passthrough replication, see also the next question.)

⇒ If I have a large network with thousands of updates taking place at all sites, won’t it take a long time for the updates to be propagated?

DataXtend RE automatically optimizes propagation times, because of its automatic load balancing and concurrent ‘passthrough’ replication ability. When a replication time occurs, sites dynamically select replication partners based on how much data they have in common and how fast the partner (and/or the network connection to the partner) can operate. Sites that contain the full database (“Complete” sites) organize themselves into an efficient tree configuration for optimal replication.

Most other replication products are very inefficient at replicating updates to large numbers of other sites.

## *Integrity and Reliability*

⇒ Does it prevent key collisions (i.e., does it allow safe distributed record creation)?

DataXtend RE’s “globally unique IDs (GUID)” are efficient 4-byte integers, and allow new records to be safely created in the same table offline at different sites without key collisions.

Some replication products offer no GUID support at all. Others support a “probably unique” ID which is a large text value (typically ~30 bytes) containing a site ID concatenated with a timestamp, a sequence number, and other information; this usually works, but it is at best inefficient.

⇒ Does it prevent all false collisions?

A false collision occurs when two, separate users, with valid update authority, simultaneously update different fields of the same record.

Resolving these conflicts is certainly a major source of administrative overhead.

DataXtend RE’s “fragments” feature correctly allows updates to different fields in the same record at different sites without any false collision. Fields that should not be allowed to change independently are defined as such by the database developer.

Most replication products do “record-level” replication, and so updates to the same record at different sites are always a collision even if the updates are to logically unrelated fields (e.g., Customer.Address and Customer.CreditRating). This is a serious integrity weakness if only one of the updates survives until an administrator checks the conflict log and reapplies the “losing” update. (Note: These “false collisions” account for the vast majority of collisions in most replication systems.)

⇒ Does it guarantee that related fields in a record will change together?

DataXtend RE’s “fragments” feature correctly guarantees that fields in the same record that should change together (as defined by the database developer) will always change together. Attempts to make competing updates to these related fields in the same record at different sites are always caught as a collision.

Some replication products do “field-level” replication, and so updates to different fields of the same record at different sites are always allowed. This is a serious integrity issue if some fields should not change independently (e.g., Customer.Address and Customer.ZipCode; if a site knows to change one, it knows enough to change the other, and since they have a “common update responsibility” the fields should change together).

⇒ How does it handle collisions?

As shown above, DataXtend RE first prevents key collisions and false collisions, which are the majority of current replication collisions. When a true collision or conflict does occur, DataXtend RE’s default behavior is ‘Most Recent Update Wins’. Although the default handling mechanism may be adequate for 99% of the situations, it can easily be changed and customized using conflict event handlers.

DataXtend RE currently differentiates between nine categories of true collisions. For example, sites A and B synch, then offline a user changes the same customer’s address differently at each site, then A and B replicate again: this is categorized as a “Type 1A”

collision because the competing updates were made based on the same pre-image (original data in that record fragment).

⇒ Does it preserve relational integrity?

DataXtend RE always automatically replicates parent records before dependent records.

Some other products do not automatically enforce correct relational integrity for foreign-key relationships.

Does it support transactions?

DataXtend RE supports full ACID transaction support without compromise to DataXtend RE's peer-to-peer, update-everywhere, and fully heterogeneous operation.

Some replication products do not support transactions. Others support transactions, but only in a "source/target," "master/slave" or "hub-and-spoke" configuration, or with only unidirectional replication. This is restrictive for users, and not scalable. Few or no other products can add transactions during replication.

## *Security*

⇒ Is it secure enough for replicating safely over public/semipublic networks like WANs, Internet and cellular connections?

DataXtend RE was designed from the start to be strongly secure. It uses industry-standard government-endorsed Triple-DES, SHA, and ECC security algorithms. All replication sessions are automatically strongly encrypted to ensure that eavesdroppers cannot read transmitted data. (This never affects performance; see the "Performance and Efficiency" section.)

Most other products do not replicate data securely. Beware products which say they will add (or have added) security in a new release: security affects a system's architecture, and it is difficult or impossible to retrofit security properly onto a system that has not been designed from the beginning with strong security in mind. If you want security, reject out of hand any product that uses "custom" or "proprietary" home-brew security; it will not be secure.

⇒ What if I do not wish to give my remote users access to my entire database?

DataXtend RE's Dynamic Data Slicing Architecture (DDSA) permits query based data deployment known as 'worksets'. This allows distribution of only the data which is pertinent to the user or site.

DDSA gives administrators 'Need to Know' security control which limits an enterprise's data exposure and controls the amount of information a user is allowed to see, without application changes.

Few or no other replication products provide these security features.

## *Ease of Use – for Developers*

⇒ Do customers have to rewrite their applications' database access code?

DataXtend RE never sits 'between' the application and the database. DataXtend RE always operates as "just another client" to the database. The application can use whatever database access method it likes.

Some other products require that the application must access the database through their proprietary ODBC driver or other fixed method. This forces the customer to rewrite the application to use that method.

⇒ Do customers have to rewrite any other part of their applications?

DataXtend RE does not require any API calls. The minimum requirement for a customer application is that the application can not change the existing primary keys.

Some other products require API calls to make the replication work. Other products are not complete runtime products at all, but only toolkits or libraries that require the customer to do actual programming.

⇒ Do customers have to make extensive changes to their database schema?

No changes need to be made to the application's schema. DataXtend RE automatically adds its own Control and System tables to the existing schema.

Some other products also do not require database changes, but instead require applications to be specially rewritten to access the database through their proprietary method. Other products require significant database changes.

⇒ Can it be completely embedded into the customer's application?

DataXtend RE's rich API can be used by any application to directly control the replication engine's configuration and functionality.

Furthermore, the DataXtend RE SDK provides administrators with the ability to create turn-key Deployment Kits, which can automatically install the replication engine, along with your custom application.

Few or no other replication products are easily embedded.

⇒ Can it be used with Java applications (or Delphi, or C++, or non-Windows, etc.)?

DataXtend RE does not sit between the client and the database, and no API calls are required, so the application can be written using any tool or language. If the application intends to make optional DataXtend RE API calls, it must only be able to call a standard Win32 DLL or use a Linux shared object.

## *Ease of Use – for Runtime Administrators*

⇒ Do administrators have to micromanage which data subsets go to which sites?

With DataXtend RE "worksets" and "fragments" features, the database designer sets up the distribution rules once, at design time. All sites, where the database is installed, will use the same shared rules. Once the replication network has been established with these rules, the Administrator can 'subscribe' individual sites to the appropriate data subsets. This can be accomplished, remotely, through the DataXtend RE Administrator application or programmatically, via the DataXtend RE API. There is no need for administrators to micromanage which data goes to which sites. Each site easy gets exactly "all and only" the subset of the database that it needs, with full referential integrity. (See the *DataXtend RE Developer's Guide* for more information.)

Other replication products require complex SQL queries to manage which data is needed at a given site. This greatly limits scalability because it is time-consuming and error-prone. (One of the most common complaints DataXtend RE has heard from customers about other replication products is that "I can't afford to put a DBA at every site.")

⇒ Can the entire database network be administered from any site, including laptops?

DataXtend RE's peer-to-peer architecture allows the entire distributed database to be administered from any site. An authorized administrator can set site replication schedules, data subset ("slice") subscriptions and any other configuration from any site. This is possible because DataXtend RE's internal housekeeping information is itself replicated using the same flexible sharing rules that DataXtend RE supports for customer databases.

⇒ Do administrators have to micromanage which sites replicate with each other?

As sites are added and removed, DataXtend RE's automatic adaptive load balancing transparently reconfigures the network dynamically for the most efficient propagation.

When a new site is installed, it performs an initial replication with any existing site, and then is a fully participating member of the database network with housekeeping information about the location, schedules, subscriptions, and other information about other sites. No manual housekeeping is needed.

Most other products require an administrator to define which sites should replicate with which others in which order. This does not scale to networks with thousands of sites.

⇒ Can it replicate through a firewall?

DataXtend RE is an official Internet standard service on TCP/IP port 242. To replicate through a firewall, simply allow port 242 inbound and outbound.

## *Ease of Use – for End Users*

⇒ Does it support true “update-everywhere”?

DataXtend RE allows database updates at any site, with no restrictions. All sites are true peers; when an update is made at one site, it is replicated to only the other sites interested in that piece of data.

Some replication products restrict updates to originate at only some sites (“master/slave,” “master/replica” and “source/target” models), or even at only one site (“publish/-subscribe” model). This can severely restrict end-user work.

⇒ Can replication be fully automated?

DataXtend RE allows full flexible replication scheduling, and authorized administrators can manage replication schedules for the entire network from any site.

Some other products cannot be fully automated without custom programming work.

⇒ Is replication transparent to the user?

DataXtend RE’s philosophy is that the user should not be able to tell the difference between working online at head office and offline on the road; the application and data are “just always there.” DataXtend RE performs replication as a background task, and users never need to know that replication is even occurring. Whenever the computer has a network connection that allows it to see another site (for example, the user has logged onto the Internet from home to browse web pages), replication will automatically occur as scheduled. If the connection is interrupted during a replication, then the next time DataXtend RE will replicate the rest of the updates with the same partner site or with any other available site. By utilizing the DataXtend RE API, much of the replication engine’s workings can be controlled by the application, so that it can become a transparent part of another software product.

Other products require users to be aware of replication, and even to initiate replication.

⇒ Can data management (“which data needs to be stored at which sites”) be made completely transparent for both administrators and end users?

Yes. DataXtend RE’s “work sets” feature makes it easy to subset the database in a way that is fully transparent to both administrators *and end users*. For example, consider a database subsetted by customer, so each site is subscribed to the information related to each customer that it cares about. Then, in the application program, when the user clicks on a customer he’s never clicked on before and the user has permission to subscribe to that customer, the application can automatically subscribe the site to that customer with a single function call. For example, if it was customer #12, the function call is just `DSECSUBSCRIBE (dbname, “Customer”, 12)`. During the next replication cycle, DataXtend RE will contact another site that already has that customer’s information and automatically add exactly and only that customer’s information into the local database; from then on, the local site is just another peer that sends and receives updates to that customer’s information. Similarly, the application can transparently unsubscribe to slices of data that are no longer needed.

Most other products require “a DBA at every site” to manage the contents of local databases. Some products allow users to manage these themselves, which reduces the need for expert DBAs but also means that replication is no longer transparent to the user.



## *About Progress Real Time Division*

The Progress Real Time Division provides event stream processing, data management, access and synchronization products to enable the real-time enterprise. Our products manage and analyze real-time event stream data for applications such as algorithmic trading and RFID; accelerate the performance of existing databases through sophisticated caching; manage and process complex data in the industry's leading object database; and support occasionally connected mobile users requiring real-time access to enterprise applications. The Progress Real Time Division is an operating unit of Progress Software Corporation (Nasdaq: PRGS), a global software industry leader. Headquartered in Bedford, Mass., they can be reached at [www.progress.com/realtime](http://www.progress.com/realtime) or +1-781-280-4000.

**PROGRESS**  
SOFTWARE

**Real Time Division**

[www.progress.com/realtime](http://www.progress.com/realtime)

### Worldwide and North American Headquarters

Progress Real Time Division, 14 Oak Park, Bedford, MA 01730 USA Tel: +1 781 280 4000

### UK Office and Northern Ireland

Progress Real Time Division, 210 Bath Road, Slough, Berkshire, SL1 3XE England Tel: +44 1753 216 300

### Central Europe

Progress Real Time Division, Konrad-Adenauer-Str. 13, 50996 Köln, Germany Tel: +49 6171 981 127

### France

Progress Real Time Division, 3 Place de Saverne, Les Renardières B, 92901 Paris la Défense Tel: +33 1 41 16 16 56

© 2005 Progress Software Corporation. All rights reserved. Progress and DataXtend RE are trademarks or registered trademarks of Progress Software Corporation, or any of its affiliates or subsidiaries, in the U.S. and other countries. Any other trademarks or service marks contained herein are the property of their respective owners. Specifications subject to change without notice. Visit [www.progress.com/realtime](http://www.progress.com/realtime) for more information.