software AG

LISTENING TO CUSTOMERS | ISSUE 3, 2009

TECHniques

Adabas
SPOTLIGHT

Experiences with Event Replicator for Adabas:
Replicating Adabas on the Mainframe to Adabas on Windows

By Dieter W. Storr, Storr Consulting, Inc

When Software AG released Event Replicator for Adabas, the company did so in response to customers' requests for the ability to have access to changed data in real-time. By listening to customers and understanding their real-world challenges, Software AG delivered a solution that proactively and efficiently delivers data updates in real-time. This article describes why organizations need replication and provides some real-life experiences with Event Replicator for Adabas.

REASONS FOR REPLICATION

In an effort to save money, many companies and state and federal agencies consider migrating their core applications to newer hardware platforms, database systems and programming languages like Oracle, SQL database, UNIX, .NET or Java. However, when these organizations perform the cost analysis, the results do not justify the expense of the migration project. Or, an organization cannot get permission from its stockholders to spend a couple of million dollars for newer technology. Additionally, there just may be no proof that such a migration project will lead to a faster process or is more secure or easier to use.

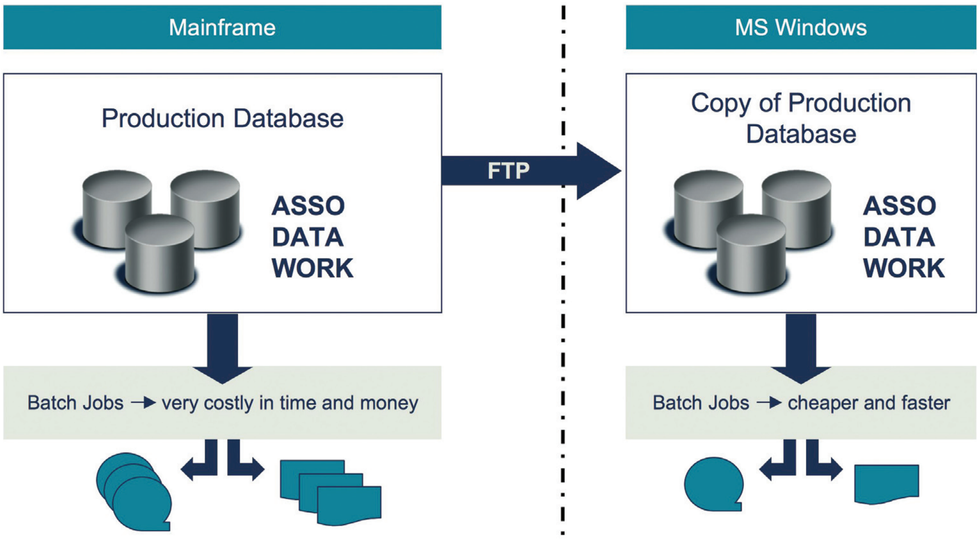
Some organizations also fail to take into account the cost of losing all of the expertise in Natural and Adabas and retraining staff members, just to be more 'sexy' in using the latest and greatest technology. In the 90s, many companies were proud to own an adaptable database system and a 4GL programming language. Programs developed on the mainframe could easily run under other hardware platforms, such as Windows, UNIX, or BS2000.

In fact, some organizations found it faster and less expensive to program in Natural under MS Windows and use Single Point of Development (SPoD) to coordinate and transport programs between the mainframe and the PC. They decompressed relevant Adabas files on the mainframe, transferred it via FTP to the PC, and compressed and re-loaded it into the Adabas database running under Windows (see Figure 1). Some batch jobs using Natural and Adabas ran on a PC up to 90% faster and cost less money than on the mainframe.

Once organizations began transporting programs between the mainframe and PC, the next challenge was figuring out how to populate the latest Adabas data faster from the mainframe to the PC in order to be up-to-date. Replication became the buzzword and some companies began to advertise that their products that could copy Adabas data by using back-ups and protection logs.

After listening to and working closely with their customers to understand their business needs in this area, Software AG offered the solution Event Replicator for Adabas to replicate data in real-time from the mainframe to the PC.

FIGURE 1: Transferring Decompressed Adabas Files via FTP to the PC


Mainframe
Production Database
ASSO
DATA
WORK
Batch Jobs → very costly in time and money
FTP
MS Windows
Copy of Production Database
ASSO
DATA
WORK
Batch Jobs → cheaper and faster

BASICS AND DEFINITIONS

For Event Replicator for Adabas, you need Adabas on the mainframe (subscription or source database) and Adabas on Windows (destination or target database). You also need a replicator engine (another Adabas database or Reptor), and on both sites, Entire Network (WCP/WCL). Subscriptions, destinations, initial-state and global format buffers can be defined online or in batch. Using the Adabas Event Replicator Subsystem (SYSRPTR) online is the easiest way; however, it can be very time-consuming to define 200 files in 50 databases online.

Start online with a couple of files in the test environment and unload the control cards into members of a partitioned data set. The batch SYSRPTR RPLOD process, with easily unloaded ADARPD cards, loads the definitions into a new Adabas system file of the Reptor database very quickly. The initial-state function is responsible for copying the first time an entire file from the source to the destination under the control of the replicator engine. At the end, activate the replication at the source database with the new ADADBS function 'replication file.'

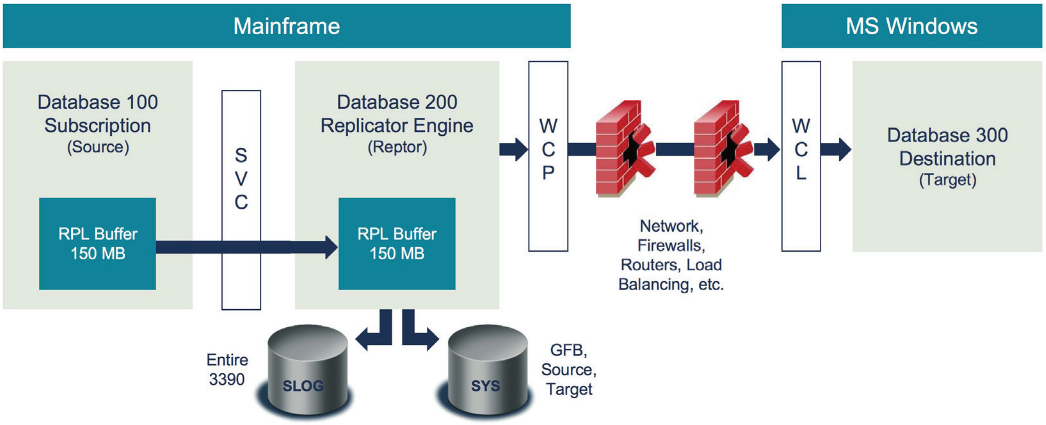
It is very important to define a large replication buffer (LRPL) on both the subscription database and replicator engine. In addition, define the SLOG file on the replicator engine in order to have an overflow area. New ADARUN parameters warn if the replication buffers reach a pre-defined high-water mark. See Figure 2.

TROUBLESHOOTING AND SOLUTIONS

During my tests with production-sized files up to 150 million records, communication problems occurred (WCP), parameters were not calculated correctly (NAB) and replication buffers (LRPL) and SLOG overflowed. Software AG's support was very helpful. Their ability to listen really helped in understanding the issues and led to correction of these values.

Another example of Software AG listening to suggestions and feedback came during some other issues that I ran into in using Event Replicator for Adabas. I ran into some communication problems on Entire Net-Work (WCP), for example RC224, RC148 SC8224, WCP0602E. I found out at the time that Software AG did not support EntireX's ability to connect Adabas

FIGURE 2: Event Replicator for Adabas



on the mainframe with Adabas on Windows. Organizations already using EntireX would also like to use it in a replication environment; however there is a little piece, an Event Replicator Target Adapter for Adabas on Windows, which is missing. It is available for other target environments, such as SQL. I discussed this with Software AG at a Users' Group meeting. Software AG listened carefully and representatives were very open to this suggestion, and R & D are currently considering this enhancement.

To double-check the replication process and compare between the source and target, I use two methods (USR4011N and add dollar fields) to compare source and target files. These procedures have helped uncover differences, and further analysis showed that some transactions received Adabas response codes from the replicator engine and were not replicated. Therefore, I wrote procedures and programs to parse and retrieve information about high-water marks of SLOG and RPL buffers, as well as response codes and warning messages.

RECOVERY AFTER A CRASH – PLOGS AND REPLAY

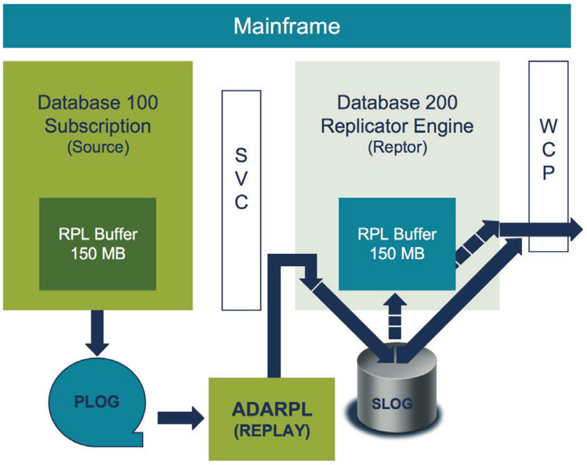
In general, Event Replicator for Adabas insures that all transactions will be replicated to the target database. However, in some cases, like an overflow of attached buffers (NAB) or replication buffers (LRPL), data may be lost and the replication must be repeated. The new ADARPL utility and its REPLAY function are similar to the known regenerate function of ADARES.

The replicator engine needs the protection log of the source database to repeat the missed transactions. The REPLAY function synchronizes the transactions from the protection log with waiting transactions in the RPL buffer and the SLOG file. See Figure 3.

CONCLUSION

Software AG was very responsive to their customers when they offered Event Replicator for Adabas. Software AG continues to listen to their customers and understand their business issues as they work to make Event Replicator for Adabas even easier to use in the Windows environment.

FIGURE 3: Recovery Process



Synchronized Mode

- Suspends new ADABAS transactions
- Reactivates files, subscriptions, destinations that are inactive
- Writes all new data to SLOG
- Synchronizes after the REPLAY new ADABAS transaction

Don't use HALT command for Reptor if data is in SLOG. A Reptor restart will delete this leftover.

To find the Software AG
office nearest you, please visit:
www.softwareag.com

SAG_ADASpot_EventReplicator_Iss3-09_18Nov09