

The Replication Mechanism in a Romanian ERP System Environment

Iuliana SCORȚA
Academy of Economic Studies Bucharest
Economy Informatics Department

Modern Relational Database Management Systems have Replication technology. Large enterprises are often spread around the country, and although WANs can be very fast and reliable these days, it is often better for each location to have a local copy of data rather than have a single database at a central location. This usually means that replication is a requirement in order for each location to have the most up-to-date data. This paper reveals a replication mechanism implemented in a Romanian ERP system environment.

Keywords: ERP, environment, system, replication.

Introduction

Replication is an architecture that allows an enterprise to maintain multiple copies of data. This is typically done to improve access to the data, by allowing various sites in the organization to have their own copies of the data. Each site then works on their own copy, and the changes made by each must automatically be reflected at any or all of the other sites.

There are different architectures which can implement replication:

- Simple copy of the database.
- Export/Import model.
- Synchronous model.
- Asynchronous model.

In synchronous model, a change in the source database is propagated to the target databases synchronously. This means that as changes occur in the source database, they appear simultaneously in all target databases. The disadvantage to this method is that if one or more of the targets is unavailable we have to use the asynchronous method.

The Asynchronous model is very popular, and is the model used by InterbaseReplicator (IBReplicator). In this model, all the data changes are stored and sent by the Replication Server to the target databases on a prior set up schedule. The advantage to this method is that if one or more of the target databases are unavailable, the data will still be replicated according to the schedule.

An organization should use the replication technology because it offers several advantages [5]:

- It Improves data availability and application performance in a distributed data environment - large enterprises are often spread around the country, and although WANs can be very fast and reliable these days, it is often better for each location to have a local copy of data rather than have a single database at a central location. This usually means that replication is a requirement in order for each location to have the most up-to-date data.
- Replication provides a backup database - large enterprises usually have sites where it is imperative to access data continuously. If a server collapses it is important to have access to the same data on a different server and this usually requires administrative intervention. Replication provides a perfectly usable backup database on another server which can be used in an instant should the primary database server fail, and can indeed be used all the time.
- For decision support systems which require near-synchronous data consistency with on-line system - usually in the decision making process it is not the on-line database that is accessed but the second database which is periodically synchronized with the first one. The replication is a solution to this problem because the data in the decision support systems is updated constantly.
- It offers consolidation of data from multiple locations - often, the decision support database contains consolidated data, and this consolidation can be done by the Interbase Replication Server, using the feature which allows replicating via a stored procedure.

Interbase replication methodology

The Interbase Replication Server consists of a Replication manager, a Replication Server and several other tools: Configuration database, Replication scheduler and Replication monitor.

The Replication Manager will run on any MS Windows machine on the network, while the Replication server will typically run on the same physical machine as the primary Interbase database, although this is not necessary - it can run on any machine on the network, as long as the Interbase client is installed.

Setting up a replication schema is fairly simple. Everything is done in the Replication Manager, and consists of the following steps:

- Define all involved databases, whether sources or targets.
- Add a Schema, and define the source database.
- Add a target database.
- Define which tables must be replicated.
- Define primary key fields, and data fields for each table.
- Create System objects on the Source Database

As soon as the system objects have been defined on the Source database, then data will be inserted into the replication log table, ready for the Replication Server to replicate. The *Replication Manager* has a number of functions [5]:

- Replication configuration tool.
 - Set up which databases are involved in replication.
 - Set up which databases are source databases, and which are targets.
 - Define Configuration parameters for each Source database.
 - Define which tables need to be replicated.
 - For each table define which fields need to be replicated. Fields can have different names and types in the source and target database.
 - Define the primary key for each table.
 - Define all usernames and encrypted passwords
 - Set up any conditions, for conditional replication.
- Performance monitoring tool that monitors

replication activity shows total number of inserts, updates, deletes, errors, conflicts encountered, creates statistics and graphics and also notifies the operator, either visually or via e-mail if errors have been encountered.

- Graphical network management tool that graphically displays all databases in replication network and depicts all links between databases and direction of data flow.
- Manual conflict resolution tool utilized to manually resolve conflicts which the server cannot do automatically by displaying all rows which failed to replicate due to conflict problems that the administrator can decide to manually replicate the row, or remove the entry.

The *Replication server* is written in the 'C' language, and uses direct Interbase API calls. This means that it can run on any supported Interbase platform, and needs no middleware (such as BDE or ODBC) to connect to Interbase. Usually the Replication Server runs as a service on NT. This allows administrators to see exactly what is going on as it happens, and also allows the administrator to initiate replication on demand.

Replication Scheduler is a utility that allows defining when replication should take place. It defines specific times or time intervals. The scheduler can be run from within the Replication manager, or separately as an NT service. The Replication Server does not need to rely on the Scheduler, as it has a built in timer, which can initiate replication on a regular basis, or replication can be initiated from any application, via an event alert.

Case study

Implementing a database replication system in a large company attracts significant costs that need to be justified by the benefits the replication mechanism would bring with its implementation. When choosing a replication oriented software we should consider several aspects [6]:

- If the application is compatible with the currently exploited database.
- Testing the application in order to see if it maps to the domain requirements of the company
- What type of replication the software ap-

plication utilizes

- If there is need of human intervention to assist the system
- The easiness in configuring the replication mechanism.

The case study is concentrated on the implementation of a replication mechanism for a Romanian ERP system that was developed using Interbase technology. In consequence the choice of mechanism is Interbase replication. The X S.A. company ERP implementation was a multi-site implementation. Also, it was imperative to access data continuously due to permanent sale activity. The IBReplicator schema view contained eight different locations from where the data was replicated in the main database. The replication mechanism is one-way type and the replication asynchronous because all of the restrictions specified by top management of the X S.A. company during the preparation phase of the ERP system implementation concerning user rights to modify automatically generated data and other hardware restrictions.

In the following I will present some considerations about the Replication methodology within Interbase with emphasis on the configuration process that took place during the Romanian ERP system implementation at X S.A. Company according to Figures 1 to 8.

In the first phase the currently exploited database structure was analyzed in order to create proper replication architecture and establish what data should be replicated and how. Because the main preoccupation was to access sales data in a continuous way the decision concerning what data should be replicated was obvious: any primary documents automatically generated by the ERP system. The structure of the database as shown in Figure 5 contains the subsequent tables: Tdoc, Tdocpoz, Tdocpozexpl, Tdocin, Tdocinpoz, Tdocinpozexpl, Tincasari, Tincasari_central_agent, Tinchiderezi, Tinventar, Tinventarpoz, Tpart. The tables contain rows of data generated within the ERP system that hold any relevancy to the sale activity for example: the Tdoc and Tdocpoz tables contain the output documents (customer invoices) and their associated positions, the

Tdocin and Tdocinpoz tables contain the input documents (supplier invoices) and their associated positions, the Tincasari tables contain the cash documents and the reference to the cashed documents, Tinventar contains inventory data, Tpart is the list of partners and Tinchiderezi contains the closing sales day information. The definition of key and data columns of these tables is shown in Figure 6. In the second phase is performed the procedure of creating the replication architecture named *clone_schema* using the IBExpert software by following the next steps:

- The registration of the replication schema database (the result are showed in Figure 1).
- Accessing the *Procedures* module and selection of the *clone_schema* procedure.
- Executing the procedure and attaching a replication schema number and committing the transaction as shown in Figure 2.

The result is a new replication schema that can be visualized with the Replication Manager and to what we can attach the databases involved in the replication process: the databases *baza* and *punct* as shown in the Figure 3. In order to maintain security the replication process is conducted via a prior set up VPN (virtual private network).

The third phase implies setting up the source and target databases involved in the replication process: the source database is *punct* and the destination database is *baza* (Figure 4). After the definition of the replication schema and the configuration of the parameters for each database involved in the replication process the *system objects* are created (Figure 5). These *object systems* are triggers and tables that help saving all the alterations made upon data in a table called *repl_log* along with the primary key of each affected row. The *system objects* are "loaded" into the system by calling the *Reload parameters* command. After this final step the replication configuration must be tested with the help of the Interbase replication server after setting the necessary parameters: database, user, password, log file and the time interval at which the replication should take place (Figure 6). If the replication is done properly the replication server will be set to automatically

(Figure 7).

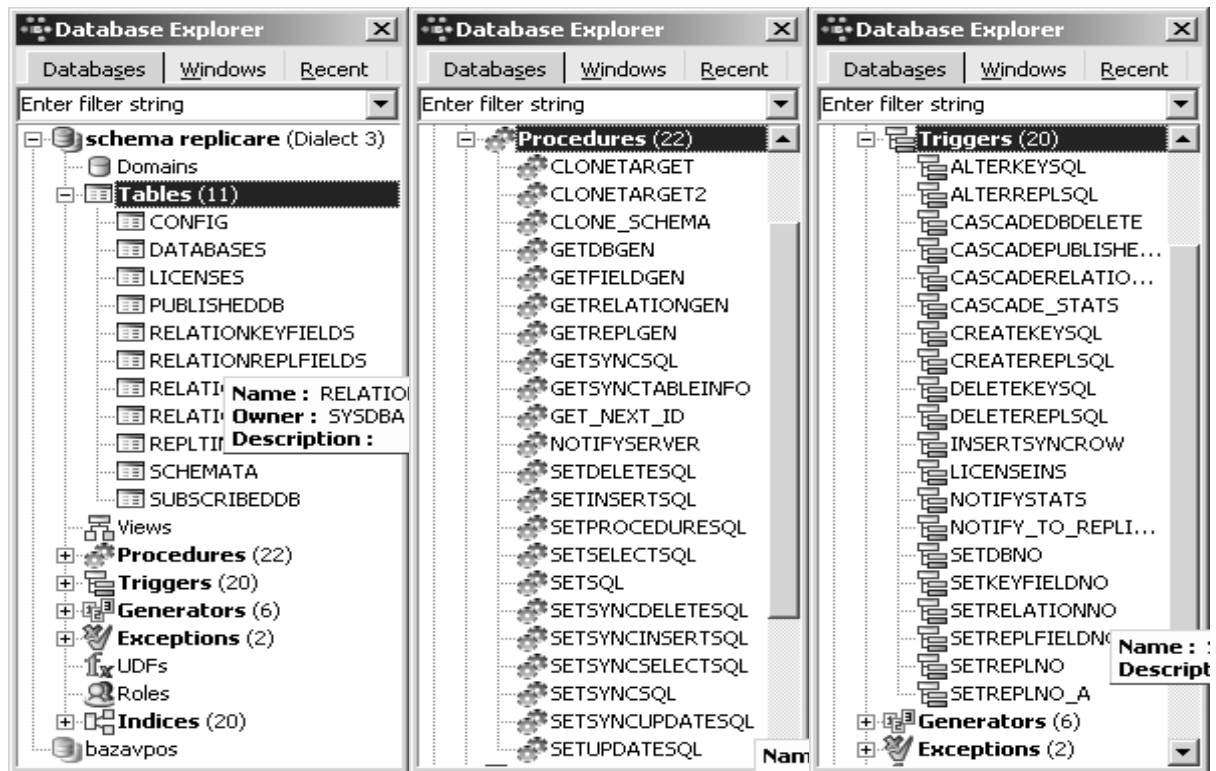


Fig.1. The structure of the replication schema database

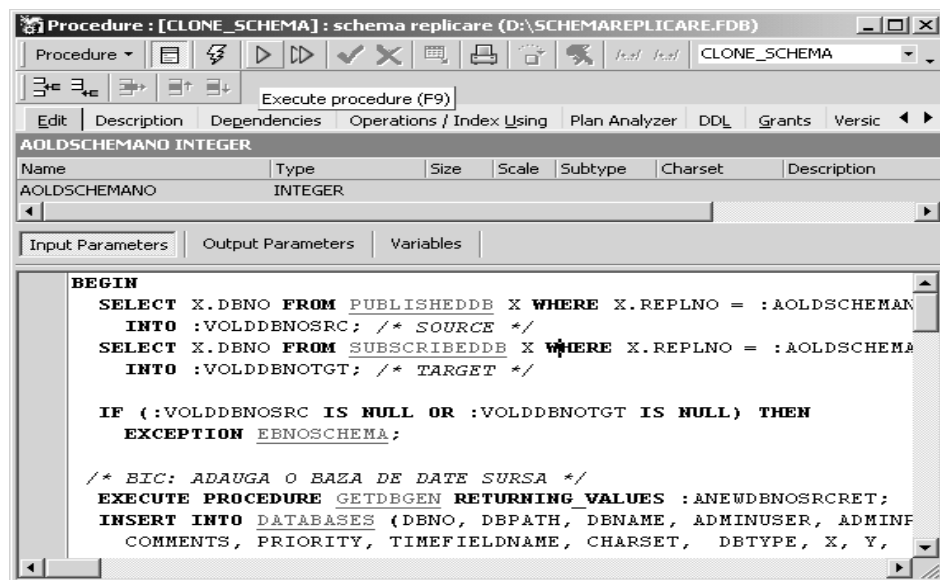


Fig.2. Executing the clone_schema procedure

In one month term after the implementation of the replication mechanism at the X S.A. company, there was registered a considerably improvement concerning the achievement of

fast and reliable sales data upon which the top managers make proper business decisions.

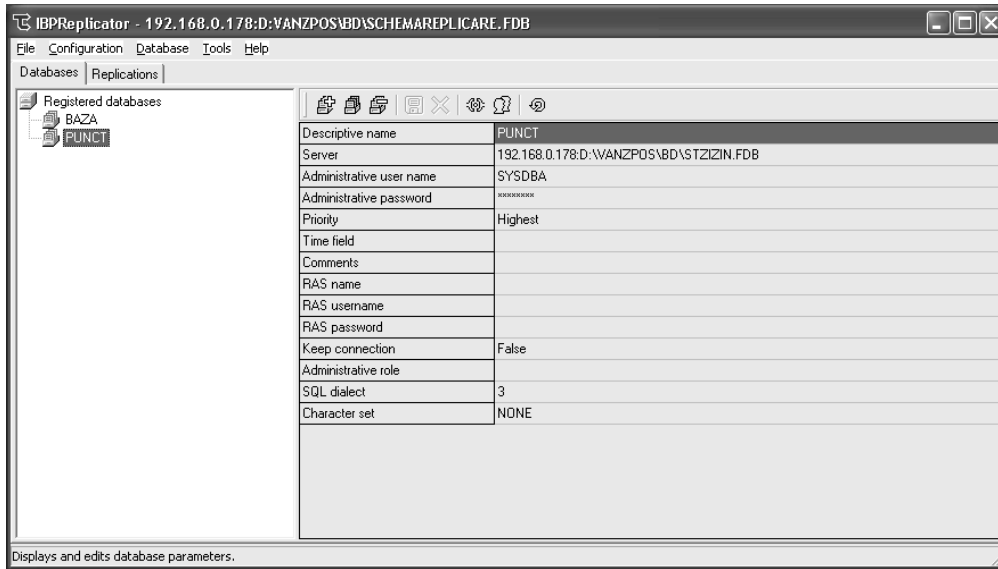
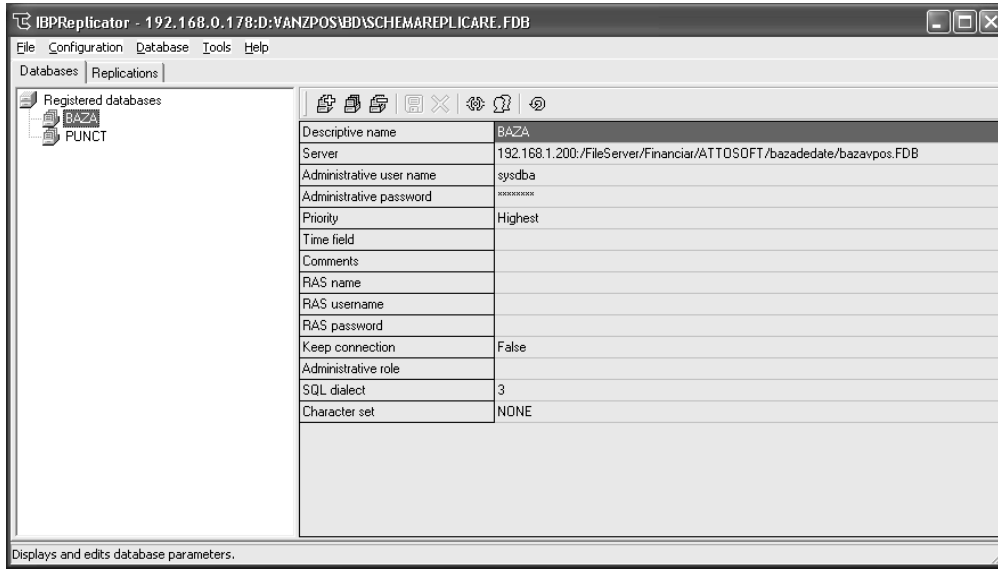


Fig.3. The configuration of databases involved in the replication process

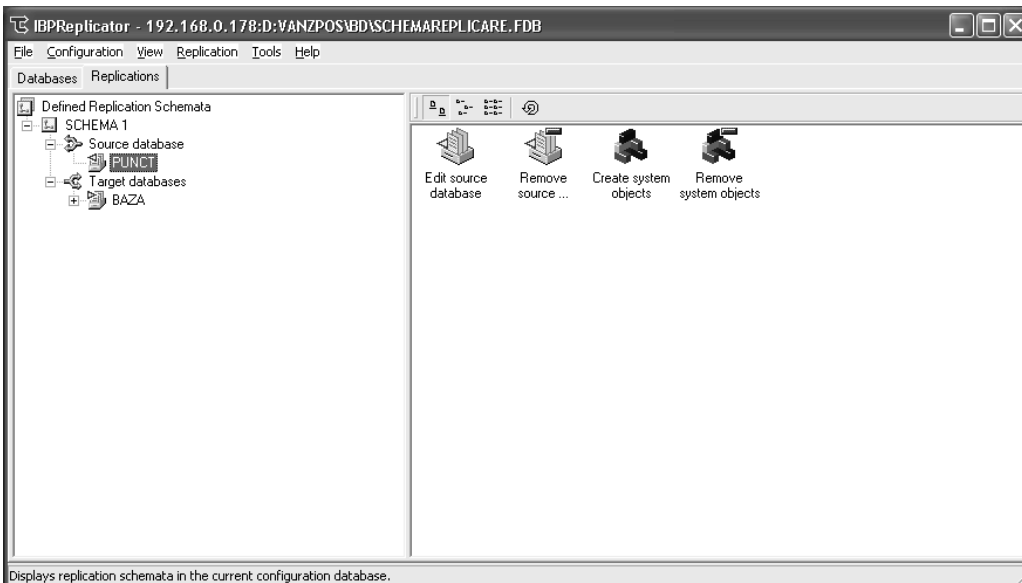


Fig.4. The configuration of the replication schema

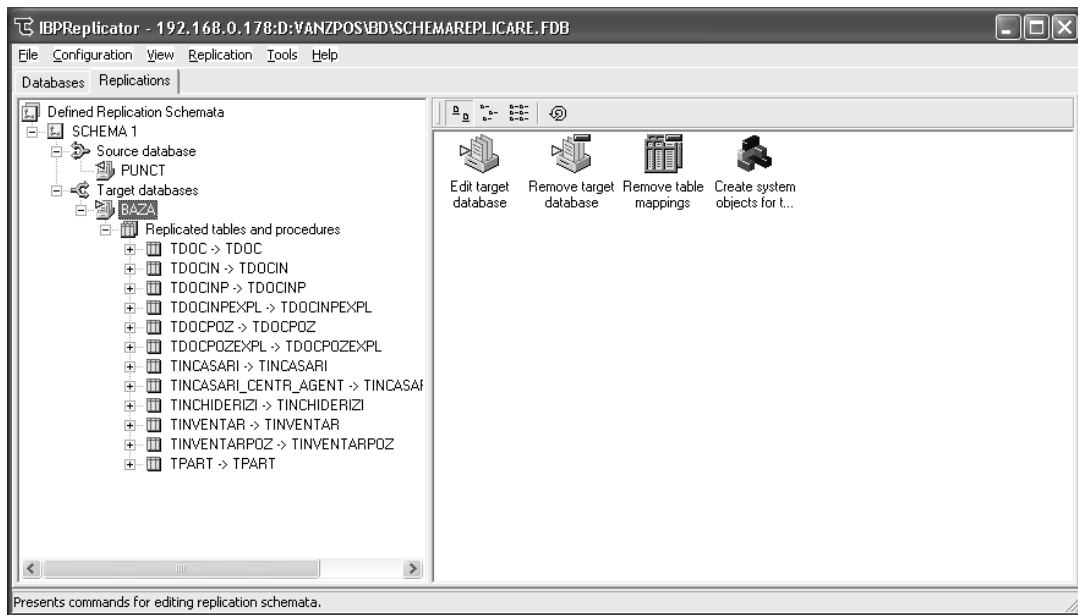


Fig.5. The configuration of parameters for each databases involved in the replication process

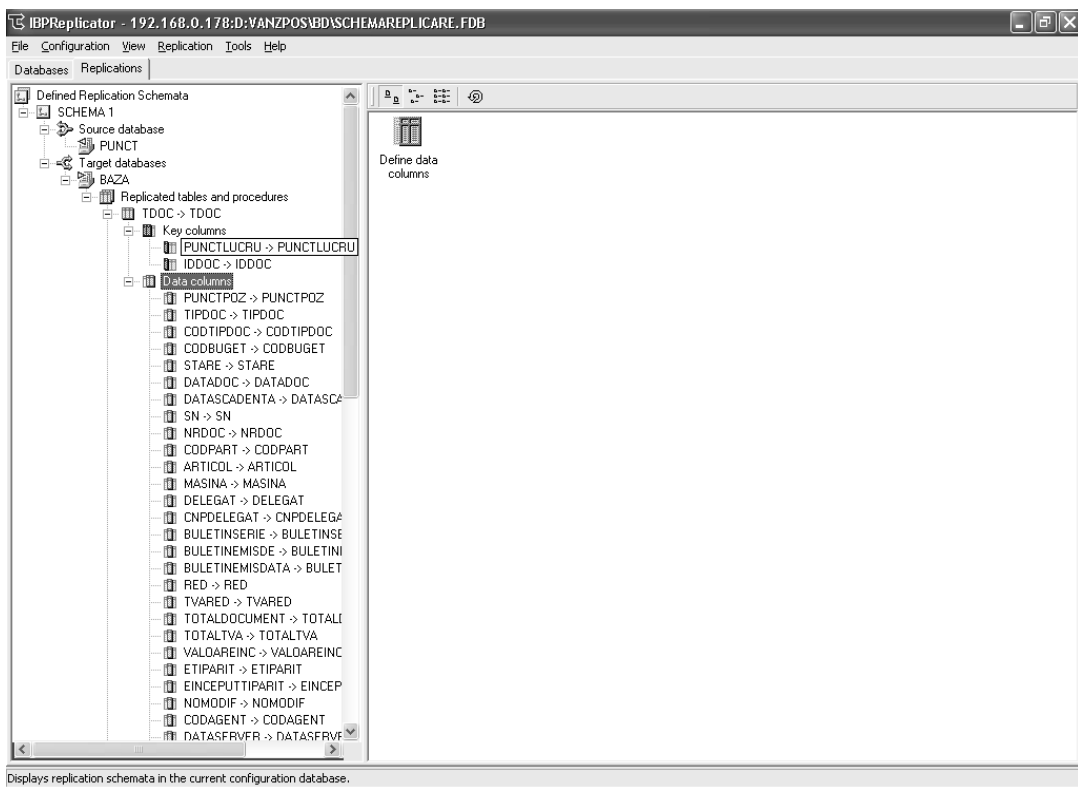


Fig.6. The configuration of fields and primary keys for each databases involved in the replication process

Conclusions

The replication technology offers the great advantage of improving data availability and application performance in a distributed data environment. Hence, the database replication provides significant importance to the efficiency of any activity developed inside an

organization. In this paper I presented the main characteristics of the Interbase replication with emphasis on the configuration of the replication process conducted in a Romanian ERP environment for automating the sales activity.

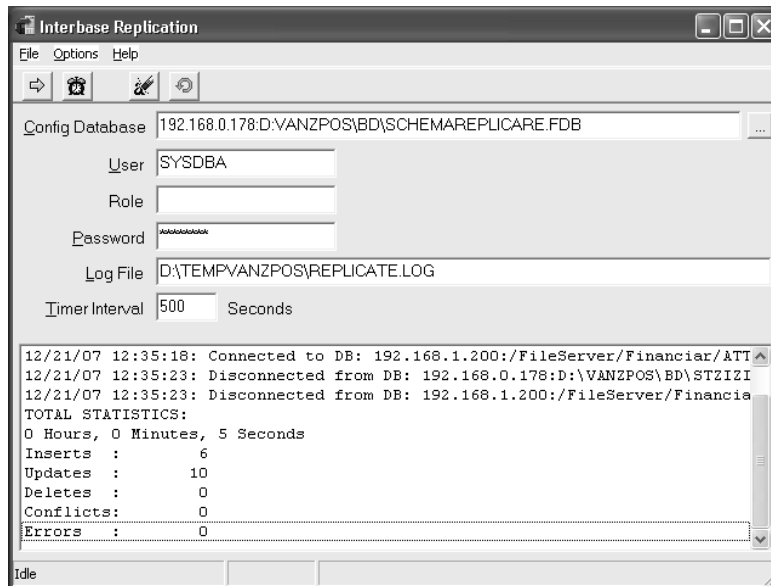


Fig.7. Testing the replication process

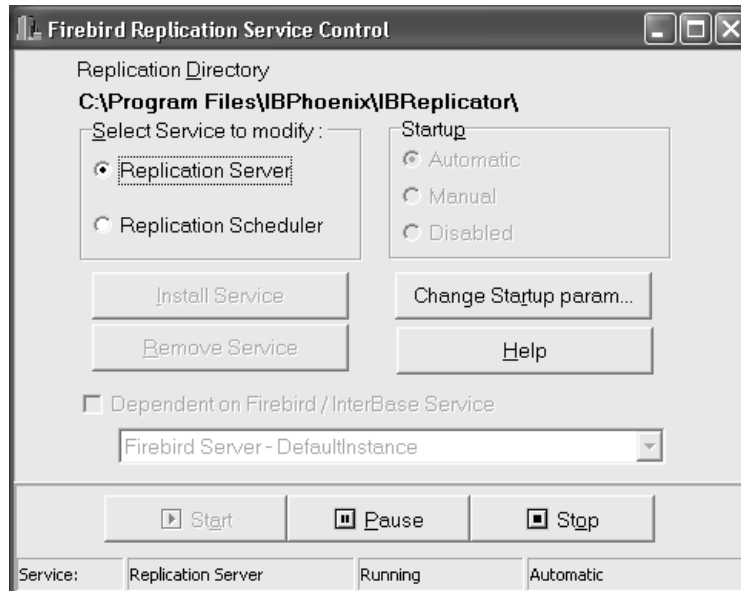


Fig.8. Firebird replication service control

Bibliography

1. IBReplicator, a Step by Step Tutorial
2. Replication: Now it's easy with InterBase V6.0 By Vince Duggan, Borland Developers Conference 1999.
3. InterBase Replication: Fundamentals By Vince Duggan, Borland Developers Conference 2000.
4. InterBase Replication: Internals by Vince Duggan, Borland Developers Conference 2000.
5. http://www.2p.cz/en/interbase_replicator/doc
6. Ion IVAN, Paul POCATILU, Marius POPA, Replicarea bazelor de date, Simpozionul "Tehnologii educaționale pe platforme electronice în învățământul ingineresc", 9 - 10 mai 2003, Universitatea Tehnică de Construcții București