

Database
Management
System

LINTER[®]

Version 5.9

Asynchronous Replication

Relational Expert Systems



Table of Contents

Introduction	4
The Asynchronous Replication Mechanism	5
Asynchronous Replication Model	6
Asynchronous Replication Component Functions	7
The Data Replication Process	8
Component Startup Order	9
Starting RELEX DB server	9
Starting the Replication Data Repository	9
Starting the Replication Server	9
Starting the RELEX DB networking agent	11
Replication Administration	12
General information	12
Creating Replication System Tables.....	12
On Win32 platform	12
On Unix/Linux platform.....	13
Replication Support System Table Structure.....	14
SERVERS table structure	14
\$\$\$REPL Table Structure	14
Replication specific SQL queries	15
Create a Replication Server.....	15
Delete a Replication Server	15
Create a Replication Rule	15
Altering a Replication Rule.....	16
Delete a Replication Rule	17
Synchronize Tables according to a Replication Rule.....	17
Replication Server Administration on Win32 platform	19
Replication Server Administration Tools.....	19
Adding a Replication Server to the Servers Table	19
Deleting a Replication Server from the Servers Table	20
Create a Replication Rule	21
Modifying a Replication Rule.....	22
Delete a Replication Rule.....	24
Browsing Replication Rules	24

Replication Server Startup/Shutdown	25
Configuring the Replication Server.....	25
Replication Server Administration on UNIX/Linux platform	26
Replication setup during RELEX DB installation	26
Use of the initial database provided with the RELEX DB distribution	27
Replication server setup for existing database	27
Create new database	27
Custom replication setup on Unix/Linux platform	27

Introduction

Many modern information systems require high performance and high availability (HA) of data retrieval in case a large number of clients operate concurrently. Besides, in the course of development, such systems should be scalable without affecting the performance.

One of the ways meeting these requirements is to create a distributed database using asynchronous data replication technique. In this case, instead of a standalone database used by all the clients of the information system, several database servers are created on different hosts of the network.

The database servers are connected to each other via network. They send each other all modifications that have been made, in order to bring all the replicated objects (tables) up to date. As replication is asynchronous, the process does not execute instantly, but takes some time during which data on different servers will not be the same. The updated data are replicated as soon as the transaction is committed and the application executed the transaction received commit response before the updated data will be replicated. This is why the replication process is asynchronous.

This approach makes it possible to significantly improve (at the best, in proportion to the number of database servers) the performance of the system, and step it up as the workload increases (e.g. the number of clients or the database size goes up) by simply adding new database servers to the information system.

To manage the system at the logical level, replication rules are used. These rules, written in the SQL database language, are descriptions of the replication procedure (i.e. what objects should be replicated, when and how).

RELEX Replication server provides the secure replication over SSL (Secure Socket Layer) TCP/IP connection optionally. RELEX Replication server allows bi-directional replication between RELEX Enterprise and RELEX Embedded Linter Series database servers.

RELEX Replication server provides one-way data replication from RELEX Enterprise/Embedded to ODBC databases. That feature is available on any RELEX platform where the ODBC manager is supported. Using that feature, the data can be replicated from RELEX Enterprise/Embedded to Oracle, DB2, MS SQL, MySQL, Sybase, PostgreSQL, etc.

The Asynchronous Replication Mechanism

To maintain the integrity of a distributed database in RELEX Embedded/Enterprise Linter Series, the mechanism of asynchronous transaction replication (hereafter, replication) is used.

The essence of the asynchronous replication mechanism is that data is processed locally, the distributed data being copied to the server on which it is to be used. This method provides the time delay then the updated data will be applied to remote host database.

If one of the distributed database servers requiring replicated data update is not available, the replication system continues working with the other database servers. The update data will be applied automatically to the database server as soon as the database servers will be available.

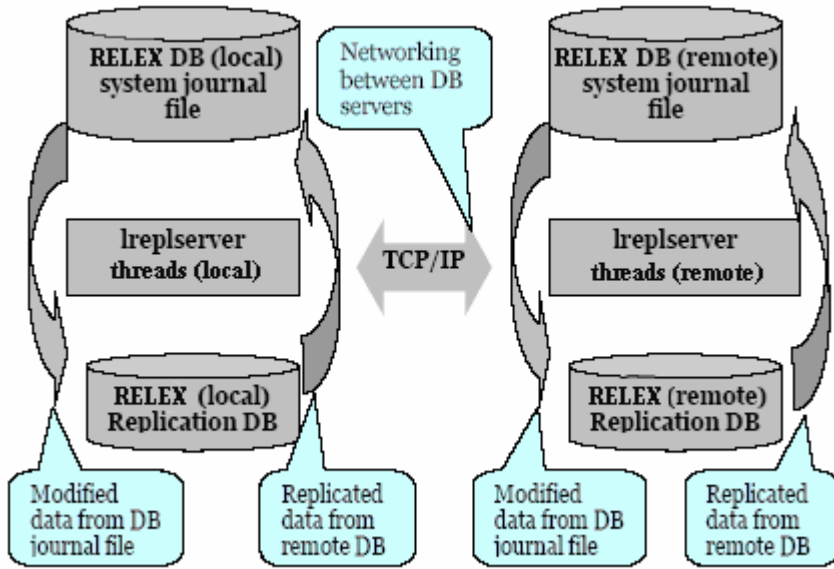
The asynchronous transactions replication mechanism guarantees that the modified data are delivered to remote database servers immediately after completion of the transaction (if the database server is available), or as soon as the remote database server is connected to the network. This approach implies storing duplicate information on different network nodes and, in comparison with other approaches towards replication, lowers the traffic, reduces the system response time. This approach provides High Availability feature for the distributed RELEX database servers.

Asynchronous replication, unlike two-phase synchronization, does not provide full synchronism of information on all the servers at any time. Synchronization takes place after a certain, usually short, time period the duration of which depends on the performance of the networking communication, database server load. For the majority of tasks, short-term presence of stale data on remote servers is perfectly acceptable.

For all that, asynchronous replication of transactions guarantees the integrity of data, as the object of exchange is the logical work unit, i.e. the transaction, not mere data from modified tables.

Asynchronous Replication Model

Two or more database servers are involved in the asynchronous replication system. The RELEX Linter Series and replication process lreplserver ensuring the implementation of the replication rules should be running on each server (see Screen 1). The replication objects are database tables, the tables menu being stored in the database (RELEX DB in Screen 1) along with the rules and database server addresses.



Screen 1 – Replication Model in RELEX Enterprise/Embedded Linter Series

The lreplserver replication server is a replication process receiving information about modified data from the RELEX DB journal files and storing it in the queues of the replication data repository. The replication data repository is an appropriately the RELEX DB instance used by lreplserver process for receiving data to be replicated and sent out.

lreplserver process spawns threads while processing replication queues data. There two types of threads: receiving thread and sending thread. That architecture allows performing replication with the maximum possible speed in concurrent mode for different remote database servers.

The asynchronous replication management in the Win32 environment is performed using the linadm system administration utility. In Unix/Linux, asynchronous replication is managed using a special batch file.

Asynchronous Replication Component Functions

The functions of the asynchronous replication system components in RELEX DB are described in Table 1.

Table 1 – Asynchronous Replication Components in RELEX DB

<u>Component</u>	<u>Functionality</u>
linter (linternt for Win32)	Main RELEX DB kernel, works independently from the other components. This component is to support only one additional function: returning modified data for the replication server.
Replication Server (lreplserver)	<p>Provides replicated database transaction tracking and records all modifications necessary for replication into the replication data repository.</p> <p>Manages the execution of the replication rules.</p> <p>Composes data to be delivered to the replication data repository.</p> <p>During the startup, starts the additional lreplserver send and receive threads.</p> <p>This component is started separately and independently from the RELEX DB server.</p>
Replication Data Repository	This is a RELEX DB instance storing data to be replicated. This data can be accessed by the replication server lreplserver threads. Also, DBA can access the replication database using RELEX DB utilities (inl, ldb, lindesk, etc.).

The Data Replication Process

A modified data delivery queue element includes complete information about the previous and current record state, the destination address, the connection number performing the operation, the transaction number and execution timestamp.

This information is entered into the delivery queue table on the replication server. Execution timestamp is used as the primary key of this table.

The `lreplserver` receive thread receives the data from the RELEX DB journal file and stores it into the reception queue table. Further on, the process composes a message notifying the sender thread that the message has been successfully received. The replication itself takes place simultaneously, the termination codes being stored in the reception queue table.

If one of the processes failed, it restarts, recovers the execution, and the replication process continues. The execution of each replication request set is tracked using the execution timestamp.

The same queues with the appropriate termination codes, operation, reception and transmission times and delivery and destination servers are used as work protocols. The monitor has access to these tables and is able to obtain any information about the replication process in real time.

If necessary, the system administrator may start the replication server queue purge procedure, and all the replicated records up to a certain date specified by the administrator will be deleted.

All exceptions of the replication process are stored in `lreplserver.log` file. That file can be generated as a set of the `lreplserver.log` file with the name of `lreplserver.log.NNN`, there NNN is the log file number.

Component Startup Order

Starting RELEX DB server

To use the RELEX DB in the replication mode:

1. The database should contain the SYSTEM.SERVERS and SYSTEM.***REPL system tables. These tables are created using the ~linter/dict/distr.sql script file.
2. The SYSTEM.SERVERS table should contain the names of the remote database servers to which the data is to be replicated. The RELEX DB database server names are entered into the table using an SQL query.

```
CREATE SERVER <server name>
```

or

```
CREATE NODE <server name>;
```

3. The ~linter/bin/nodetab file used by the lreplserver send and receive threads should also contain the names of the remote database servers to which the data is to be replicated.

Starting the Replication Data Repository

The following should be started as a replication data repository:

- 1) in the UNIX environment:
 - a RELEX DB instance (with a different to RELEX DB replicated instance value of the LINTER_MBX shell environment variable);
- 2) in the Microsoft Windows NT environment:
 - a RELEX DB instance the option /name=XXXX key.

By default, the repository database name should be REPLSERV. If it is not the case, the database name should be implicitly indicated when the replication server is started (see section “Starting the Replication Server”).

There should be meta-tables existed in the replication data repository database. These tables are created during the replication server installation using the arepl.sql file.

Starting the Replication Server

The replication server is the lreplserver program.

Lreplserver command line startup options are following (for Win32 and Unix/Linux):

<u>Option</u>	<u>Description</u>
-u <name>/<password>	User name and user password of the replicated database.
-us <name>/<password>	Replication data repository user name and password.
-t <number>	Replication server message tracing level (value from 0 to 4), 0 is brief format, 4 is maximum detailed format.

<u>Option</u>	<u>Description</u>
-s <repository database name>	Replication data repository database name (if differ to the regular REPLSERV name).
-clear <clearing mode>	Clean the replication data repository. Available modes: <ul style="list-style-type: none"> • ALL – clean the data of the entire replication database; • NNN – clean all processed replication data within interval of time NNN, there NNN is the number of second.
-log TRACELEVEL <number>	Log file creation. The <number> sets the tracing level of messages placed into the log file.
-nohist	Clean all processed data.
-p <tcp/ip port number>	TCP/IP port number.
-l <name>	The database instance name/ID.
-stop	Remove the replicated database checkpoint set by the replication server lreplserver.
-down	Shutdown the replication server.
-cfg =<file specification>	Sets the name of the text file containing the lreplserver startup options.
-?/?	Display lreplserver startup options.
-n	The path to the nodetab file directory. By default, the directory of lreplserver file is used.
-odbcstorage DSN	Database Source Name for the one way replication to the ODBC database.

Startup options available only for Win32 platform:

<u>Option</u>	<u>Description</u>
-d	Debugging mode (debugging information will be displayed to the console).

Startup options available only for Unix/Linux platform:

<u>Option</u>	<u>Description</u>
-daemon	Start lreplserver as a Unix daemon.
-ssl	SSL protocol should be used for secure communication over TCP/IP.

Examples

```
lreplserver -U=SYSTEM/MANAGER -US=SYSTEM/MANAGER
-s=REPLSERV -daemon -p=1080 -t=3 -log=5
```

The replication server starts the replication send and receive threads automatically.



To search for the startup directory, lreplserver uses the PATH environment variable.

Starting the RELEX DB networking agent

The `dbc_tcp` RELEX DB networking agent is started on a replicated database in the regular way.

Replication Administration

General information

To be able to replicate data from the replicated database to other remote databases, the RELEX DB replication server should have the following information:

- 1) Which specific user tables should be replicated;
- 2) What remote databases the user tables are to be replicated to. This information is set using the so-called replication rules. The replication rules are created and stored in the replicated database, in the database system table SYSTEM. \$\$\$REPL. Besides that, the replicated database should contain the SYSTEM.SERVERS system table listing all remote RELEX DB servers that the replication rules refer to. RELEX DB server names in the SYSTEM.SERVERS table should be the same as the RELEX DB server names in the ~linter/bin/nodetab file used by the RELEX DB server on this local host to access remote database servers. The communication protocol for the remote database servers referred by ~linter/bin/nodetab file should be ATCPIP, which means asynchronous replication protocol. For the SSL support the ATCPIPS should be used.

Creating Replication System Tables

Replication system tables are created automatically during the replication server installation (if the corresponding option is set - see "Installing RELEX DB Replication Server"). If this option is not set during the installation, the replication system tables can be created manually.


On Win32 platform

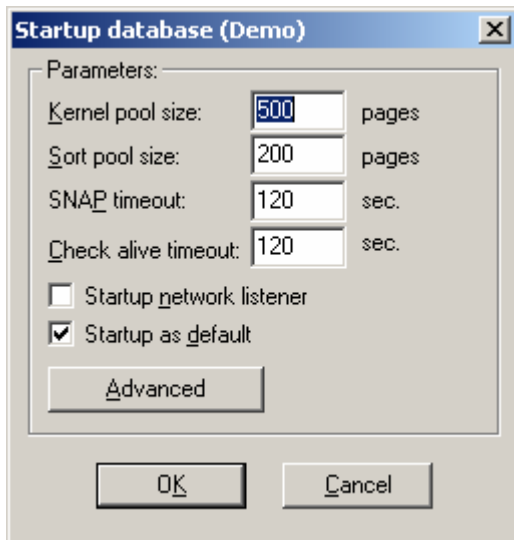
To create replication system tables:

1. Execute the linadm utility as a service.
2. Expand the server tree and select the source database of the RELEX DB replication server for which the replication meta tables are to be created (Screen 2).



Screen 2 – Selecting the source database to create replication meta tables

3. Execute the **Database > Startup ...** command, or hit F9, or click on the  icon, or right-click the mouse and select Startup from the drop-down menu. The database startup dialog box will pop up (see Screen 3).



Screen 3 – RELEX DB server startup parameters dialog box

4. Set the Start network listener flag (if the tables are created in the remote source database), or the Start as default flag (for the local database).
5. Click OK.
6. Return to the RELEX DB server tree and make sure that the database has been started.
7. On the local host, start the RELEX DB networking agent for the selected remote database (if the tables are being created on a remote database).
8. Make sure that the source database of the replication server is accessible from the local computer.
9. Execute the MS Windows command for starting the inl utility: **Start > Programs > RELEX Enterprise Linter Series > Interactive SQL.**
10. In answer to the inl prompt, enter, for instance, `SQL>d:\Program Files\linter\bin\dict\distr.sql`.
11. Check that the tables have been created:

```
SQL>show SERVERS
SQL>show $$$REPL
SQL>exit
```

On Unix/Linux platform

To create replication system tables:

1. Go to the directory where the replicated database is located, e.g.:
`[sh]$cd ~/linter/db`
2. Start RELEX DB instance if it is not running:
`[sh]$../bin/linter`
3. Start the Linter command interface program:
`[sh]$ ~/linter/bin/inl -u SYSTEM/MANAGER`
4. In answer to the inl hint, enter the path to the distr.swl replication table creation command file, e.g.:

```
SQL>_/usr/linter/bin/dict/distr.sql
```

5. Make sure that the tables have been created:

```
SQL>show SERVERS
SQL>show $$$REPL
SQL>exit
```

Replication Support System Table Structure

SERVERS table structure

The SERVERS table is used to store the information on all Linter SQL servers to which the data is replicated from the replication server source database. As mentioned above, the names of Linter SQL servers in the SERVERS table should be the same as the Linter SQL server names in the nodetab file that uses the client network driver started on the replication server source database.

SERVERS Table Structure:

<u>Column</u>	<u>Description</u>
server_id	Linter SQL server ID. Filled in by the Linter server.
flags	Reserved for future use.
servername	Unique Linter server name (no longer than 8 characters). The value is assigned by the system administrator when adding a record.
last_connect	Date and time of the last connection of the replication server with this Linter server. Filled in by the Linter SQL server.

\$\$\$REPL Table Structure

The \$\$\$REPL table is used to store the replication rules (instructions on which tables of the source database are to be replicated, and to which Linter SQL servers). Each source database may have its own set of replication rules.

\$\$\$REPL table structure:

<u>Column</u>	<u>Description</u>
REPL_ID	Replication rule system ID. Filled in by Linter SQL server.
REPL_NAME	Unique name of the replication rule (no longer than 66 characters). The value is assigned by the administrator when creating the replication rule.
DST_SERVER	System ID of the server to which the table is to be replicated (SERVER.SERVER_ID column).
SRC_ID	System ID of the replicated table (\$\$\$SYSRL.\$\$\$S11 column).
DST_OWNER	Replicated table owner ID (no longer than 66 characters).
DST_TABLE	Replicated table name (up to 66 characters).
DST_PWD	Replicated table owner password (up to 18 characters).
STATE	System information about the replicated table state.

Replication specific SQL queries

Create a Replication Server

Adds a replication server to the replication servers system table.

Syntax

```
<Create replication server>::=CREATE SERVER <Server name>;
```

Description

- 1) <Server name> is an identifier no longer than 8 characters.

Remarks

- 1) The replication source database should contain the SYSTEM.SERVERS system table.
- 2) <Server name> should reference with one of the RELEX DB server names from the ~linter/bin/nodetab file of the RELEX DB with the ATCPIP or ATCPIPS communication protocols.
- 3) To create a replication server, it is necessary to have the replicated database DBA rights.

Delete a Replication Server

Deletes a replication server from the replication servers system table.

Syntax

```
<delete replication server>::= DROP SERVER <DB Server name>;
```

Description

- 1) <DB Server name> is an identifier no longer than 8 characters.

Remarks

- 1) <DB Server name> should be listed in the SYSTEM.SERVERS system table.
- 2) To delete a replication server, it is necessary to have the replicated database DBA rights.
- 3) All replication rules referring to the deleting database server will be removed as well.

Create a Replication Rule

Creates a replication rule and adds that information into the SYSTEM.\$\$\$REPL system table.

Syntax

```
<create replication rule>::=
CREATE REPLICATION RULE <Rule_name>
FOR <Source_table_name>
TO <Receiver_table_name>
ON SERVER <Server_name>
USER <User_name>
[PASSWORD '<Password>' ]
[ENABLE | DISABLE ]
[ SYNC | ASYNC ]
[IGNORE OLD VALUE|CHECK OLD VALUE|CORRECT NUMBERS ];
```

 CHECK OLD VALUE and CORRECT NUMBERS are reserved or future use.

Description

- 1) <Rule_name> is an identifier that should be unique in the database and not exceed 66 characters.
- 2) <Source_table_name> and <receiver_table_name> set the names of user database tables. View names and system table names cannot be used.
- 3) <Server_name> sets the name of the RELEX DB server, there the receiver table is located.
- 4) <User_name> is the remote database user name.
- 5) <Password> is the remote database user password.
- 6) The default values are ENABLE, ASYNC and IGNORE OLD VALUE.

Remarks

- 1) The replication source database should include the SYSTEM.SERVERS and SYSTEM.\$\$\$REPL system tables.
- 2) The schema of the source table should be identical to that of the receiver table.
- 3) Replication of tables containing columns with Extfile data type is not allowed.
- 4) Both the source table and the receiver table should have a column with a PRIMARY KEY attribute.
- 5) A replication rule may be created by the owner of the replicated table, or by a user having the SELECT, DELETE, INSERT and UPDATE privileges to this table.
- 6) ENABLE makes the rule available, while DISABLE makes the rule unavailable.
- 7) SYNC sets the synchronous replication mode, while ASYNC sets the asynchronous replication mode. In the asynchronous mode, data replication is performed as far as possible.

 In the current version of the RELEX DB, the SYNC option is unavailable.

- 8) Replication modes in case of conflicting record values in the databases:
 - IGNORE OLD VALUE – ignore the inconsistency of the old value;
 - CHECK OLD VALUE – check the old value and return an error if the values do not fully coincide;
 - CORRECT NUMBERS – if numeric values do not coincide, preserve the difference between the old value and the new one.

Altering a Replication Rule

Alters the properties of an existing replication rule.

Syntax

```
<alter replication rule>::=
ALTER REPLICATION RULE <Rule_name>
[PASSWORD '<Password>' ]
[ENABLE | DISABLE]
[SYNC | ASYNC]
[IGNORE OLD VALUE|CHECK OLD VALUE|CORRECT NUMBERS ];
```

Description

- 1) The <Rule_name> identifier should be unique in the database and not exceed 66 characters.
- 2) <Password> is the remote database user password.
- 3) The default values are ENABLE, ASYNC and IGNORE OLD VALUE.

Remarks

- 1) <Rule_name> should refer to a rule already existing in the \$\$\$REPL table.
- 2) A replication rule may be altered by the replicated table owner or by a user having the SELECT, DELETE, INSERT and UPDATE privileges to this table.
- 3) The values of the ENABLE, DISABLE, SYNC, ASYNC, IGNORE OLD VALUE, CHECK OLD VALUE and CORRECT NUMBERS parameters can be found in Section "Create a Replication Rule".

Delete a Replication Rule

Deletes an existing replication rule.

Syntax

```
<delete replication rule>::=  
DROP REPLICATION RULE <Rule_name>;
```

Description

- 1) <Rule_name> is an identifier that should be unique in the database and not exceed 66 characters.

Remarks

- 1) <Rule_name> should refer to a rule that already exists in the SYSTEM.\$\$\$REPL table.
- 2) A rule may be deleted by the replicated table owner or by a user having the SELECT, DELETE, INSERT and UPDATE privileges to this table.

Synchronize Tables according to a Replication Rule

Synchronizes the data of source table and remote table. This query is typically used after creating a new replication rule. Before starting to use t, the source table and the remote table involved in the replication should be synchronized.

Syntax

```
<table synchronization>::=  
SYNCHRONIZE REPLICATION RULE <Rule_name>;
```

Description

- 1) The <Rule_name> identifier should be unique in the database had not exceed 66 characters.

Remarks

- 1) <Rule_name> should refer to a replication rule already existing in the \$\$REPL table.

- 2) Tables may be synchronized by the replicated table owner or by a user having the SELECT, DELETE, INSERT and UPDATE privileges.


Replication Server Administration on Win32 platform

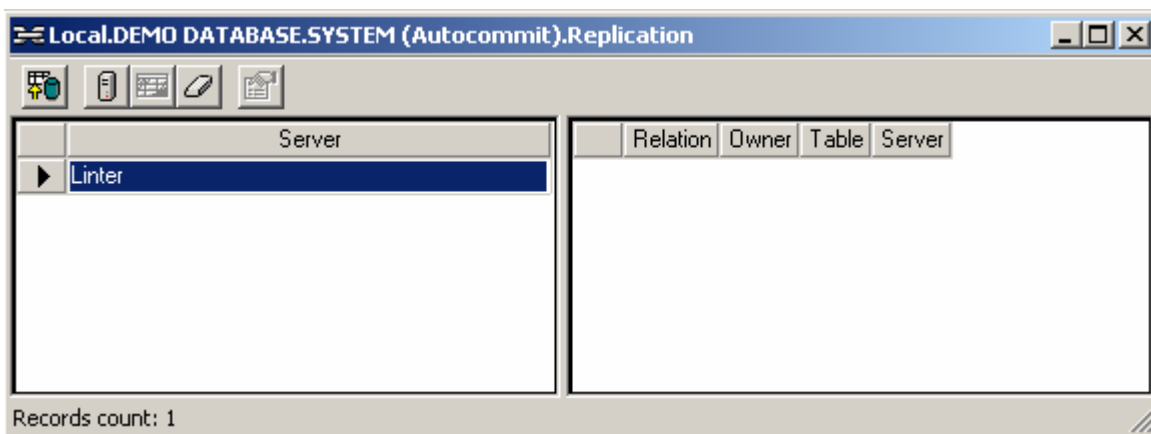
Replication Server Administration Tools

Replication server administration may be carried out in one of the following ways:

- 1) using the command interface (see Section “Creating Replication System Tables” for information how to start it);
- 2) using the lindex utility (Linter desktop).

To start this utility:

1. Execute the linadm utility in the RELEX Enterprise/EmbeddedLinter Series server of the source database (see Section 24.6.2).
2. Start the RELEX DB networking agent.
3. Execute the Linter desktop utility from linadm. To do this, go to the linadm window and execute the **Tools > Desktop command**, or press Ctrl+D, or click on the () icon.
4. In lindex, execute a connection with the replication source database.
5. Execute the **Database > Replications** command in lindex. This command generates a dialog box for entering information in the SYSTEM.SERVERS and SYSTEM.***REPL tables (Screen 4).



Screen 4 – Replication objects management dialog

Adding a Replication Server to the Servers Table

There are two ways of adding records to the SERVERS table:

- 1) using the command interface;
- 2) using the Linter desktop utility.

To add a record using the command interface:

1. Start the command interface (see section “Creating Replication System Tables”).
2. In answer to the inl hint, enter the command:

```
SQL>create server <RELEX DB server name>
```



```
(<RELEX DB server name> is the name of the newly added RELEX DB server, e.g:
SQL>create server sale;
SQL>create server "Market$";
SQL>create server "HumanResources";
3) Repeat Operation 2 for all servers added to the table.
```

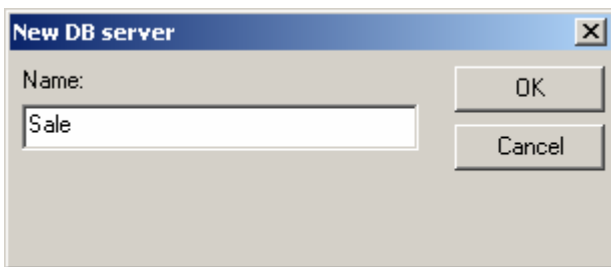
4) Exit from the inl utility:

```
SQL>exit
```

 A server may as well be added (using an SQL query) to the SYSTEM.SERVERS table using linsk (see "Linter Desktop", Section "Executng SQL Queries").

To add a record using the Linter desktop:

1. Execute the linsk utility (see section "Replication Server Administration Tools").
2. Activate the SERVERS table frame (click the mouse on the left frame). The **Replication > New server...** menu item and the () icon will activate.
3. In the linsk utility window, execute the **Replication > New server ...** command, or press Ctrl+Ins, or click on the () icon.
4. Enter the new server name in the window that appears (Screen 5).
5. Click OK to confirm the operation. To cancel, click Cancel.
6. Repeat the operations for each server to be added.



Screen 5 – Server addition dialog box

Deleting a Replication Server from the Servers Table

There are two ways of deleting records from the SERVERS table:

- 1) using the command interface;
- 2) using the Linter desktop utility.

To delete a record using the command interface:


1. Execute the command interface (see section "Creating Replication System Tables").
2. In answer to the inl hint, enter the command:


```
SQL>drop server <RELEX DB server name>;
(<RELEX DB server name> is the name of the RELEX DB server to be deleted,
e.g.:
SQL>drop server sale;
SQL>drop server "Market$";
SQL>drop server "HumanResources";
```

3. Repeat Operation 2 for all servers to be deleted from the table.
4. Exit from the inl utility:

```
SQL>exit
```

To delete a server using the Linter desktop:

1. Execute the lindesk utility (see section “Replication Server Administration Tools”).
2. In the lindesk utility window, execute the **Replication > Delete server** command, or press Ctrl+Del, or click on the  icon.
3. In the standard confirmation dialog box, click OK or Cancel.
4. Repeat Operations 1-3 to remove other servers.

 You cannot directly rename a server in the Servers table. To rename a server, delete the server with the old name and add a server with a new name.

Create a Replication Rule

There are two ways to create a replication rule:

- 1) using the command interface;
- 2) using the Linter desktop utility.

To create a replication rule using the command interface:

1. Create a text file, e.g. d:\repl\repl_rule.sql, containing the text of a replication rule (or several replication rules) to be created. The server names that the replication rule refers to should be listed in the SYSTEM.SERVERS system table, e.g.:

```
CREATE REPLICATION RULE repl_table1
FOR table1 TO table1
ON SERVER sale
USER Manager;
CREATE REPLICATION RULE repl_table2
FOR table2 TO table2
ON SERVER sale
USER Manager
IGNORE OLD VALUE;
...


```

2. Execute the command interface (see section “Creating Replication System Tables”).
3. In answer to the inl hint, enter the file execution command, e.g.:



```
SQL>_d:\repl\repl_rule.sql
```

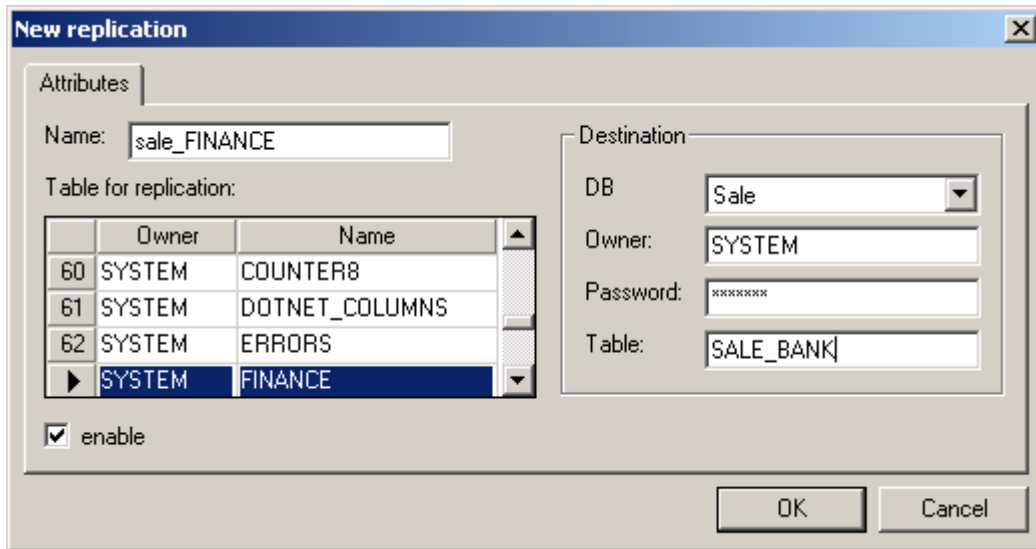
4. Terminate the inl utility:

```
SQL>exit
```

 Replication rule creation using an SQL query may as well be performed in lindesk (see "Linter Desktop", Section "Executing SQL Queries").

To create a replication rule using the Linter Desktop utility:

1. Start lindesk (see section “Replication Server Administration Tools”).
2. Activate the \$\$\$REPL table frame (click the mouse on the right frame). This operation activates the **Replication > New relation...** menu item and the  icon.
3. In the lindesk window, execute the **Replication > New replication...** command, or press Ctrl+Alt+Ins, or click on the  icon. The replication rule creation dialog box will pop up (Screen 6).



Screen 6 – Replication rule creation dialog box

4. Fill in the fields in the dialog box (see Table 2).

Table 2 – Fields and items of the replication rule creation dialog box

<u>Field</u>	<u>Description</u>
Name	Replication rule name (unique in the source database) – a string no longer than 66 characters.
Table for replicating	Table to be replicated (selected from the complete list of source database tables).
Destination	Remote database parameters.
Server	Remote database RELEX DB server name.
Owner	Remote table owner name.
Password	Remote table owner password.
Table	Remote table name.
Enable	Replication rule state flag: if set, the rule will be used; if not, the rule is disabled.

5. Click OK to confirm the replication rule creation. To cancel, click Cancel.

 To create a replication rule with additional options not listed in the dialog box, use the command interface.

Modifying a Replication Rule

There are two ways of modifying a replication rule:

- 1) using the command interface;
- 2) using the Linter Desktop utility.


To modify a replication rule using the command interface:

1. Create a text file, e.g. d:\repl\alter_rule.sql, containing the text of the replication rule (or rules) to be modified. The name of the replication rule referred to in the ALTER REPLICATION... SQL operator should be listed in the \$\$\$REPL system table, e.g.:



```
ALTER REPLICATION RULE repl_table1 DISABLE:
ALTER REPLICATION RULE repl_table2
IGNORE OLD VALUE;
...
```

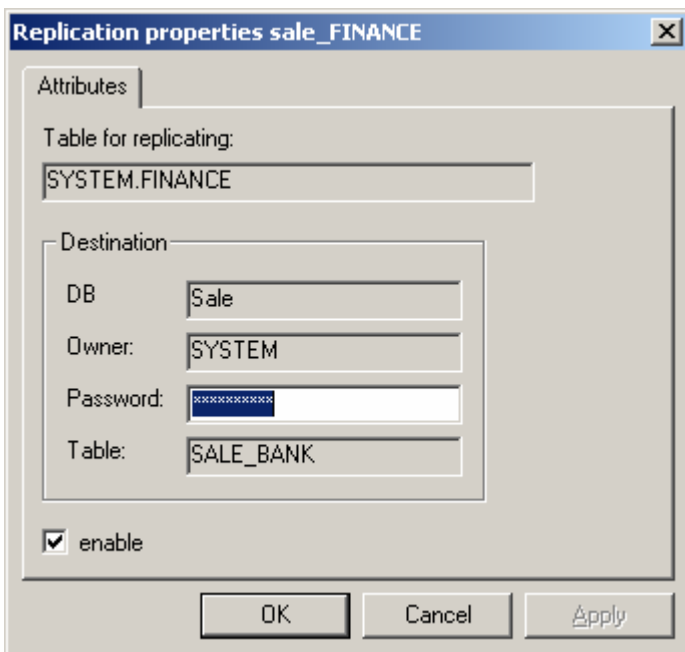
2. Start the command interface (see section "Creating Replication System Tables").
3. In answer to the inl hint, enter the file execution command, e.g. SQL>_d:\repl\alter_rule.sql
4. Exit from the inl utility:

```
SQL>exit
```

 Replication rule modification using an SQL query may as well be performed in Indesk (see "Lintor Desktop", Section "Executing SQL Queries").

To modify a replication rule using lodesk:

1. Start the lodesk utility.
2. Activate the \$\$\$REPL table frame (click the mouse on the right frame). This action enables the **Replication > New relation...** menu item and the  icon.
3. In the lodesk utility window, execute the **Replication > Properties...** command, or press Alt+Enter, or click on the  icon. The replication rule modification window will be displayed (Screen 7).



Screen 7 – Replication rule modification dialog box

4. Fill in the replication rule modification window fields (see Table 3).
5. To confirm the operation, click Apply, and then OK. To cancel and close the window, click Cancel.

Table 3 – Fields and items of the replication rule modification dialog box

<u>Field</u>	<u>Description</u>
Password	Remote table owner password. String no longer than 18 characters. During rule modification, the value is not checked for correctness (an incorrect value will be detected only when the rule is applied by the replication server).
Enable	Replication rule state flag: if set, the rule will be executed; if not, the rule is disabled.

Delete a Replication Rule

There are two ways of deleting a replication rule:



- 1) using the command interface;
- 2) using the Linter Desktop utility.

To delete a replication rule using the command interface:

1. Start the command interface (see section “Creating Replication System Tables”).
2. In answer to the inl hint, enter the rule deletion command, e.g.:

```
SQL> drop replication rule "Repl_Rule_Bank";
```

To delete a replication rule using the Linter Desktop utility:

1. Start lindesk (see section “Replication Server Administration Tools”).
2. Activate the \$\$\$REPL table frame (click the mouse on the right frame). This action enables the **Replication > Delete menu** item and the  icon.
3. Select the rule from the replication table frame.
4. In the lindesk utility window, execute the **Replication > Delete command**, or press Ctrl+Del, or click on the  icon.
5. Click OK in the standard window to confirm the operation. To cancel the deletion, click Cancel.

Browsing Replication Rules

There are two ways to browse a replication rule:



- 1) using the command interface;
- 2) using the Linter Desktop utility.

To browse a replication rule using the command interface:

1. Start the command interface (see section “Creating Replication System Tables”).
2. In answer to the inl hint, enter the rule deletion command, e.g.:

```
SQL> select * from SYSTEM.$$$REPL
1> where rule_name='Repl_Rule_Bank';
```

To browse a replication rule using lindex:

1. Start lindex (see section "Replication Server Administration Tools").
2. Activate the \$\$\$REPL table frame (click the mouse on the right frame). This operation activates the **Replication > Properties...** menu item and the  icon.
3. Select the rule from the replication table frame.
4. In the lindex utility window, execute the **Replication > Properties...** command, or hit Alt+Enter, or click on the  icon. A dialog box with additional replication rule options will appear.
5. To browse the rule, click OK.

Replication Server Startup/Shutdown

The replication server startup/shutdown procedure is described in "Linter SQL Server Administrator for Windows NT", Section 7.9.

Configuring the Replication Server

The replication server configuration procedure is described in "RELEX DB Server Administrator for Windows NT".

Replication Server Administration on UNIX/Linux platform

To setup replication process the database servers should be accessible over TCP/IP networking.

Replication setup during RELEX DB installation

To setup replication process the replication feature should be included into RELEX DB distribution. RELEX DB distribution contains installation batch file `install.sh`.


During `install.sh` batch file execution the following question will be asked:

Create Linter start procedure script?

The answer should Y. After that the installation script will prompt to use existing demo database, supplied with the RELEX DB distribution, or to create a new one. For new database creation the prompt or the database path will be asked:

Enter the database directory:

The full directory path should be entered (root directory/'included) as a database path. If the database directory doesn't exist, `install.sh` script will prompt for the database directory creation.

 The database directory is created using 'mkdir' command, so recursive directory name is not allowed.

The database will be created promptly.

For the replication database the following question will be displayed:

Create auxiliary database needed for asynchronous replication feature activation?

If the replication database is not required, the answer should be 'N', otherwise the replication database will be created in database/storage directory.

The next question will prompt user to create RELEX DB startup/shutdown shell scripts:

Create Linter start/stop procedure for the database?

To create the scripts reply Y.


For replication server start, please answer Y on following question:

Do you want to activate asynchronous replication feature?

The installation procedure will prompt for the remote database server properties: arbitrary name, TCP/IP address. The <ENTER> response on remote replication server name will finish the replication servers properties setup.

The result of the `install.sh` procedure execution will be:

- Newly created replicated database;
- Newly creation replication database in storage directory;
- Replication server startup and shutdown scripts located in `~linter/bin` directory: `async_start.sh`, `async_stop.sh`.

 Scripts `async_start.sh`, `async_stop.sh` are used by other RELEX DB startup scripts.

To start/stop replication server at the replicated database server startup/shutdown time the following scripts should be used: `start_linter.sh` and `stop_linter.sh`. These scripts are located in `~linter/bin` directory.

Use of the initial database provided with the RELEX DB distribution

The initial database is suitable for most of the cases, especially for enterprise installations.

For the embedded system setup the installation setup should be modified. The modifications should be done in `gendb` command (`~linter/db/make_db.gdb` file and `~linter/dict/storage.gdb` file) in order to create more economical database in terms of the file system space usage.

Replication server setup for existing database

In case, the system already contains existing RELEX DB database without replication server installed, the new scripts can be created using `make_scripts` batch file. During the execution the following questions will be prompted:

- existing database directory name;
- create a replication database;
- create remote database server names with IP-addresses;
- `make_scripts` batch file creates startup/shutdown scripts as well.

Create new database

To create new database, please use the `~linter/bin/create_database` shell scripts. The newly created database will be identical to the initial database provided in RELEX DB distribution.



The startup/shutdown scripts are designed to support only one instance of replicated database and one instance of the replication database. It is possible to run concurrently on the same host several instances of replication (source) database and several instances of the replication server. The setup procedure in case of multiple database instances is more complicated and requires accurate work.

Custom replication setup on Unix/Linux platform



That information is for DBA users.

Below the replication server setup steps are listed.

1. Create replicated (source) database using `gendb` utility. Execute `sql-scrpt` `~linter/dict/distr.sql` against created database. Restart RELEX DB server.
2. Create replication database executing `~linter/db/make_storage`. The shell script creates the database in current directory. Before executing `make_storage` script, change the current directory to the replication database directory. The database can be created without `make-storage` usage. To do so, please execute following steps:
 - Create database using `gendb` utility;
 - Setup RELEX DB instance ID (shell command `LINTER_MBX=NNNN`, there `NNN` is the RELEX DB instance ID);

- Start RELEX DB kernel (linter) against replication database and execute `~linter/dict/arepl.sql` script using `inl` utility. Restart linter process;
 - Unset `LINTER_MBX` environment variable.
3. Edit file `~linter/bin/nodeab`:
- Add new line to the file: `'REPLSERV local 12345'`, there `REPLSERV` is an arbitrary name for the replication server database, `local` is a reserved keyword, `12345` is a database instance ID;
 - Add new line `'REMOTEDB ATCPIP my_name.my_domain 1080'`, there `REMOTEDB` is an arbitrary name for the remote replication server. `ATCPIP` is a communication protocol for replication, `my_name.my_domain` is an ip-address of the remote replication server, and `1080` is a TCP/IP port of the remote replication server. If the local database doesn't replicate the data back to the remote database, then the information in `nodetab` file is not necessary.
4. Create new replication server, using `inl` utility:
- ```
inl -U SYSTEM/MANAGER
sql> create server REMOTEDB;
```
5. Start `dbc_tcp` (RELEX DB networking agent).
6. Start replication server `lreplserver`. The command line or `lreplserver` should provide:
- a username and password for the DBA of the source (replicated) database and replication database;
  - Replication database name (`REPLSERVER` is a default);
  - Path for `nodetab` file;
  - TCP/IP port;
  - `'-daemon'` option;
  - trace level for the `lreplserver.log` file.
7. For the source (replicated) database replication rules should be created for the table to be replicated (create replicaion rule.. ).