

Database
Management
System

LINTER[®]

Version 5.9

Troubleshooting Guide

Relational Expert Systems



Software License and Limited Warranty

The Linter software to which this document refers is licensed under the terms of a separate contract. That contract also describes the limited warranty.

Publication Notice

RELEX has done its best to insure the accuracy of the material in his documents. However, please be aware that the document may contain errors and that enhancements to the software may make parts of the document inaccurate. RELEX does not guarantee the accuracy of the information in this document.

Copyright and Contact

Copyright 1999-2006 Relex, Inc.

119, 20-letia Oktyabrya St., Voronezh, 394006, Russia.

Phone: (+7-4732)711-711

Trademark Notices

IBM	DB2		
Ingres	Ingres		
Informix	Informix		
Microsoft Corp.	Windows 95/98/NT	FoxPro	
Oracle Corp.	Oracle		
Relex, Inc.	data river	inl	lindesk
	dbstore	kom	linter
	gendb	lhb	

Table of content

Software License and Limited Warranty	2
Publication Notice	2
Copyright and Contact	2
Trademark Notices	2
Introduction	4
Error reporting.....	4
RELEX database result codes	4
Linter.out file.....	4
LINTER.LOG file.....	6
LINTER.LOG record tags	8
Application lock troubleshooting.....	9
Linter kernel internal information	9
Database consistency verification	9
Networking troubleshooting.....	10

Introduction

RELEX database offers a full set of the troubleshooting methods: log files, test utilities, etc. The document contains the information for DBA, application developers and end users.

Error reporting

To report a problem to RELEX Inc. support team, please use RELEX Ticket Tracking system. In order to access RELEX Ticket Tracking system the Internet connection should be available and one of the Internet browsers. RELEX TT registered users shall login and enter the ticket. Users, who have no account at RELEX Ticket Tracking, please use button 'Send Ticket' to generate the ticket report form.

The following field of the ticket form should be filled in:

- Valid e-mail address, in order to receive the response Product name, product component and product version Ticket title and description.
- In case there is additional information for ticket, such as database archive, log-files, etc. please attaché those files as a zip-archive to the ticket - 'Attach' field.
- The 'External ticket' field will help make a reference to external defect tacking system, which contains the related to the filled ticket information. That field will be used for future reference.

RELEX database result codes

Every query/request to RELEX database server receives the response code and the second result code, which contains the operating system error code - errno. The 'errno' code is OS dependent result code and can be found in OS documentation or correspondent errno.h file of the OS SDK.

For example, RELEX database message in case the database is not found:

```
16.05.2003 23:46:04 E-501 (OPFIL):  
Could not open file.\\1.01, system error 2
```

Here 501 is the RELEX I/O error code, which can be found in the list of the RELEX database result codes. The 'system error 2' is the operating system code, which is ENOENT.

The complimentary error code from operating system shows the real reason why the file is not found: besides the trivial wrong database path, it can be permissions problem - 35, etc.

All RELEX result codes are listed in -linter/dict/errors.lod file.

Linter.out file

The 'linter.out' file is created by Linter kernel in the database directory.

That file contains:

- The startup/shutdown information, so DBA can very the startup/shutdown events;
- All database exceptions will be logged into the linter.out file. The result of the 'GET LINTER DUMP' command.

Each record in linter.out file contains the timestamp of the message. That timestamp is by Greenwich time.

Here is the example of the linter.out startup records:

```
16.05.2003 13:10:50 RDBMS Linter SQL v.5.9.23.9 connected to
database "DEMO DATABASE "
16.05.200313:10:50 POOL holds 880 pages
16.05.2003 13:10:50 Table queue size : 100
16.05.2003 13:10:50 Column queue size : 503
16.05.2003 13:10:50 Channel queue size : 103
16.05.2003 13:10:51 File queue size 30
16.05.2003 13:10:51 User queue size : 100
16.05.200313:10:51 Procedures table is absent
16.05.200313:10:51 Procedures Dictionary table is absent
16.05.200313:10:51 Security groups table is absent
16.05.200313:10:51 Security levels table is absent
16.05.200313:10:51 Devices table is absent
16.05.200313:10:51 Statons table is absent
16.05.200313:10:51 Replcaton rules table is absent
16.05.200313:10:51 Trgger table is absent
16.05.200313:10:51 Filer table is absent
16.05.200313:10:51 File extension table is absent
16.05.200313:10:51 Sequence table is absent
16.05.200313:10:51 Errors table is absent
16.05.200313:10:51 In-kernel backup table is absent
16.05.200313:10:51 Audiing disabled
16.05.200313:10:52 Transaction contol is turned on
16.05.2003 13:10:52 Sort queue size : 1
Existing contol point(s) in database:
Contol point lst is empty.
16.05.2003 13:10:53 Kernel system parameter: MBX - "13498", Pid -
3328
Copyright (C) 1995-2006 Relex, Inc. All rghts eserved.
16.05.2003 13:10:53
16.05.2003 13:10:53 Fail! NO LICENSE FOUND!
16.05.2003 13:10:53 *** RDBMS Linter is running
16.05.2003 13:10:53 *** Press <ENTER> for shell prompt
```

As shown, Linter startup records contain Linter startup parameters and the license information. Also, startup messages contain Linter kernel process id (pid) information which related to the running on that database Linter instance - Pid - 3328 and the Linter instance number - MBX - "13498". The Linter instance identified on Unix-like platforms as a arbitrary number (default is 20561), and on Windows machines this is an instance name, like "Demo", which has correspondent record in ~linter\bin\nodetab file.

The linter.out file contains error events, like I/O operaton error, here is the example the database fe open error:

```
17.05.2003 23:41:14 E-501 (OPFIL):
Could not open file e:\Program files\Linter\Db\Demo\23.01,
system error 35
```

Here is E-501 points to the Linter result code 501 and an errno = 35, which can be find in OS manual or SDK include-file errno.h content.

In case, Linter kernel was not shutdown property, the linter.out file will contain full information regarding Linter journal recovery process.

The information for required indices, which was identified by Linter kernel (/autoindex Linter startup option), will be also located in linter.out file.

In case, user needs to report database failures or other situation, which should be addressed by RELEX support team, the file linter.out is very useful to be presented to have the situation analyzed in details.

LINTER.LOG file

One of the very useful information as well as for development and as for support is the information stored in LINTER.LOG file. The LINTER.LOG file is supported by RELEX Enterprise version, the RELEX Embedded supports LINTER.LOG feature only by customer request. LINTER.LOG file contains all the requests Linter processed in order they have been received.

LINTER. LOG might be requested by RELEX support team in order to identify the problem. RELEX also developed player utility for LINTER.LOG file, which allows simulating reported problems.

LINTER. LOG file can be created by the user request. There are several ways to activate LINTER.LOG file creation:

- 1) At Linter startup procedure. For Unix-like platform, there are options for Linter startup - /LOG /LOGALL. /LOG option forces Linter kernel to log all received from application requests. /LOGALL has the same action, but in addition, the timestamp for each record will be wren to the LINTER.LOG file. On Windows platform to turn LINTER.LOG records on, the menu should be used.
- 2) By SQL query. The query can be sent by user application or by inl utility, for example. The SQL Query is 'set log on' or 'set log full', 'set log on' query forces to start logging all received by Linter kernel requests to the LINTER.LOG file, while 'set log full' query requests the query timestamp be logged to the LINTER.LOG file as well.

In order to execute 'set log' query, user needs to have 'DBA' permissions.

To turn off the LINTER.LOG records the query 'set log off' can be sent, or Linter kernel should be restarted.

Linter kernel doesn't overwrite LINTER.LOG file, all new records will be appended to the existing LINTER.LOG file. In case, user needs to erase LINTER.LOG file, the logging process should be stopped and LINTER.LOG file can be removed securely.

How is the LINTER.LOG can help? Let imagine, there is a promoted application, which is running on Linter database and the error 2202 is discovered while the application were executing the queries. The LINTER.LOG file will contain a query text, like:

```
! :E=0 :T=04 :37 :07 .260 :C=2 :
?TRAN:T=04 :37 :37 .995 :C=2 :L=65535 :P=0 :
select * from employees;
! :E=2202 @&#:O=1507329 :T=04 :37 :38 .005 :C=2 :L=65535 :
```

Here is the string with 'E=2202' shows the Linter result code for he query 'select \$ from employees' is 2202, which is 'table not found'. So, based on diagnostic DBA can adjust the database or submit sufficient details on application diagnostics.

In case, of application poor performance troubleshooting, LINTER.LOG file is very helpful. Let consider an example:

```
?SLCT:T=17 :35 :53 .978 :C=2 :L=65535 :P=0 :
```

```
select * from auto where personid in (select personid from person
where firstnam like 'F%');
!:E=0:T=17:35:54.218:C=2:A=14:
```

The query start time is 17:35:53.978 and the result were return by Linter to application on 7:35:54.218, so the query execution time is 240 milliseconds. As a result DBA can very the database schema and optimized without tuning application or DBA may submit the required information to application developers.

Analyzing LINTER.LOG file it is important to remember that query execution is a concurrent processing, so the query response doesn't necessary follows the query records. To recognize the correspondent lines of the LINTER.LOG file the tag 'C=...' shall be used. That tag is the Linter channel number, which is used for query execution Database connections or cursors are associated with the Linter channels.

Database query record begins with the '?' character, while the query response begins within '!' character.

In case the LINTER.LOG record begins with '#', it means the query was spawned by Linter kernel.

During the query execution, which involves database stored procedure or trigger calls, the additional Linter channel might be opened. Here is an example of query trace:

```
update person set personid=2600 where personid=600;
```

The existing trgger called during the query execution:

```
trigger "AUTO_UPDATE" after update on "SYSTEM"."PERSON"
for each row old as "OLD" new as "NEW"
execute FOR DEBUG
code
execute direct " update auto set personid=" +
ittoa(new.personid) +" where personid = " + itoa(old.personid) +
";";
end;
```

LINTER.LOG records	Comments
?OPEN:T=18:45:18.476:U="SYSTEM"/"MANAGER":P=0:R=0:	Received 'OPEN' command, initiating to open database connection.
!:E=0:T=18:45:18.476:C=2:	Returned Linter channel number is 2.
? :T=18:45:43.622:C=2:L=65535:P=0:	Received 'update' query
update person set personid=2600 where personid=600;	Query text
?OCUR:T=18:45:43.622:C=2:P=0 :R=0:	Linter spawns open cursor for trigger execution
#? :T=18:45:43.622:C=3:	Execute trigger query
update auto set personid=2600 where personid = 600;	Trigger query text
#!:E=0:T=18:45:43.622:C=3:A=1:	Trigger query executed with no errors
!:E=0:T=18:45:43.622:C=2:A=1:	Return 'update' query result – no errors
?CLOS:T=18:45:45.905:C=2:	Received 'CLOS' command from application

LINTER.LOG record tags

LINTERLOG file records begin within '?' or '!' or '#' characters. After the string beginning character the Linter call-interface command following or the SQL query text or LINTER.LOG record tag. Below the LINTER.LOG records abbreviation listed.

LINTER.LOG file lexeme	Explanation
:C=nnn	Channel number
:T	Record timestamp
:E=nnn	Request result code
:L=nnnnn	Row buffer size
:P=nnnn	PRZEXE field of the Call interface control block
:O=nnnn	Operating system result code
:I=nnnn	RowID field of the Call-interface control block
:J=nnnn	Application process id
:N=cccc	Linter database server name
:K=nnnn	Number of rows field of Call-interface control block
:A=nnnn	The Control block KolKor parameter for GETM command

Application lock troubleshooting

To verify the status of some query within the database server, Linter kernel provides such an information by the SYSTEM.\$\$\$CHAN table content. The \$\$\$CHAN table columns are described in details Linter dictionary document'. There is a column 'LOCKED_BY'. The column contains the reference to the channel number, which holds required resources. Let assume there are two processes with the pids 2345 and 6789 are running and process #6789 is suddenly stopped. The following select query result gives an idea why the process is hanging:

```
select channel, status, client_pid, locked_by
from $$$chan where client_pid in (2345, 6789);
```

CHANNEL	STATUS	CLIENT_PID	LOCKED_BY
34	IDLE	2345	0
45	BUSY	6789	56
56	IDLE	2345	0

The query result displayed that channel #45, which was opened by process #6789 is waiting for channel #56, which were opened by process #2345. Unfortunately, the existing version of Linter doesn't allow to display the database object which is the locked channel waiting for - that feature will be added in next Linter release.

Linter kernel internal information

RELEX database design provides a full set of features allowing troubleshooting process to be performed without visiting customers site or login to a remote customer's machine. One of the 'last resort' in some of very complicated situations can be the full dump of the Linter kernel without stopping Linter kernel execution. The Linter dump can be received by the executing 'get linter dump' command by inl utility or by any other database interface available for user. In case, Linter kernel is hanging, on Unix-like systems to initiate the 'get linter dump' query, DBA can send SIGUSR2 to linter process, that can be done by 'kill' command from the OS shell command prompt:

```
kill -17 <linter pid>.
```

At the moment 'get linter dump' request execution, Linter will suspend execution of all other requests until the 'get linter dump' command will be done. The Linter dump information can be found in linter.out file. To obtain the explanation of the situation, the linter.out file shall be submitted to RELEX support team.

Database consistency verification

The reason then user might need to verify the database consistency can be very various:

- The thorough system verification is needed;
- Before backup the database, user needs to be sure that the database is 'good'.
- Some unexpected system behavior is happening etc.

There are two ways to verify the database consistency:

- 1) on-line, without bringing down Linter kernel and allowing to continue queries processing.

- 2) off-line. The Linter kernel should be shutdown and the utility testdb may be used for database consistency verification and repair if needed. On-line database consistency verification. To perform the verification inl utility can be used:

```
INL> test table <table name>;
```

As a result the table consistency report is generated, 'test table' query locks the table for the access while the verification is in progress. To restore the database in on-line mode the queries 'press table' and 'rebuild able' can be used.

Off-line database consistency verification. For the off-line database consistency verification process, Linter kernel shall be stopped or the database, which needs to be verified. Testdb utility should be used or the verification The testdb options are described in details in 'test utility' document. There are several hints might be helpful:

- 'testdb' command creates the database consistency report;
- 'testdb -r' command restores the database consistency for most of the cases;
- 'testdb -r -il' command restores the database consistency in case the Linter journal wasn't finished or lost;
- 'testdb -r -s upd.sql' command restores the database and creates the 'upd.sql' sql script with the index restore queries. File upd.sql shall be executed after Linter will be restarted.

To very the Linter backup file consistency, there is an lhb utility option '-t', which allows verifying database backup file consistency verification.

Networking troubleshooting

The networking communication in Linter is done in a very simple fashion, so there are not a lot of parameters involved into the networking.

- 1) On database server side.

DBA needs to be sure:

- Linter is running and accessible by the local applications;
- The Linter server listener (dbs) is running and the networking port opened by dbs is opened for the remote machine.

- 2) On client side.

DBA needs to be sure:

- Linter networking agent (dbc) is up and running;
- The remote server is accessible via network;
- The remote database server parameters in nodetab are valid: database name is in upper case and length is not more then 8 characters, the networking protocol is valid (TCPIP is more popular protocol), the remote machine address is valid as well as the port.