

Database
Management
System **LINTER**®

Version 5.9

Data Base Structure Testing

Relational Expert Systems



Table of contents

Introduction.....3

Requirements.....4

Program Launch5

Command Line Format 5

Program Options and Arguments 5

Program Execution7

Testdb Messages8

Introduction

The testdb program is used to test the physical structure of Linter databases. It runs diagnostics on the internal integrity of the database as and checks for incompatibilities among data elements, testdb does not test databases with fatal structure faults. If the Linter kernel cannot launch a database, the testdb program will not run either.

Requirements

This program functions only when the Linter kernel is inactive. During the course of its execution, testdb produces diagnostic messages that can be routed to a file. Analyzing these messages the DB administrator can find faulty database sections and determine whether they will influence the process of working with the database and find solutions to fixing the problem. Most of the problems may be fixed by using the testdb program with the -r option. After the errors have been corrected, the testdb program must be run once more to insure that the DB structure has been corrected. If the database structure could not be restored it must be restored from the last archived copy.

Program Launch

Command Line Format

The testdb command line format is the following:

```
testdb [<options>] [<database_path>]
```

Program Options and Arguments

The <database_path> argument is the path to the database to be tested. If this argument is omitted, testdb will look for the SY00 operating system environment variable and check that directory. If SY00 has not been defined, the DB is assumed to be in the current directory. See document "Installation and Basic configuration" for the SY00 variable.

The following options and their arguments are available to the testdb:

<u>Option and Argument</u>	<u>Description</u>
-de	deletes dispensable files from database directory. Used only in combination with -r option.
-df	Deletes invalid references. Used only in combination with -s option.
-di	Delete indexes. Used only in combination with -r, -t options.
-dr	Deletes invalid rows. Used only in combination with -r option.
-f <diagnostics file>	assigns a file to hold duplicates of all the diagnostic messages produced during DB testing. The default is no file.
-g	creates a file with a .gdb extension for storing commands to correct errors using the gendb utility.
-h	displays help info on testdb and its options.
-I <index testing level>	determines extent of index testing. Values can range from 1 (test only the index structure) to 3 (perform in-depth testing of the indexes). For large DBs, index testing is usually the most time consuming process. The default value is 2.
-il	ignores the unfinished log.
-l	generate the list of the table names.
-m <tracing level>	determines the level about which, and therefore the quantity of, tracing messages that are displayed on screen by the testdb program. Messages from all levels prior to <tracing level> are displayed. The valid range is 1-5: 1 – only main stage messages: the beginning and end of testing; 2 – info on objects, tables, being tested; 3 – info on elements (data files, index files, BLOB data files) of objects being tested; 4 – info on index files;

<u>Option and Argument</u>	<u>Description</u>
	5 –info on the number of tested entries in the object. Large DB testing requires substantial time. It is advisable to assign the value 5 to <tracing level> to be able to tell whether the program crashed or hung. The default value is 2.
-o	Controls the operation of the testdb.
-p<pool size>	sets the size of the internal pool, which is used during the DB testing. Its value is analogous to the pool size set during kernel launch. The more RAM the machine has the bigger the pool can be and therefore the faster testdb will work. The default size of the pool is 500 pages @ 4KB.
-r	automatic repair mode: all structural faults that can be repaired are repaired based on info contained within the DB itself.
-s<file name>	<file name> is the name of a file that will contain SQL-operators for rebuilding incorrect indexes and other damaged structures. These SQL queries may be used later to rebuild damaged DB fragments using, for example, the inl program.
-t<table name>	Test only the table specified by <table name> instead of the entire DB.
-tn	If this option is set, NULL will be returned instead of error or zero in a Date data type.
-ts	Tests only system tables and integrity constraint conditions.
-tz	Errors in a DATE data type return NULL by default. If this option is set, zero will be returned instead.
-u	User/password.

Program Execution

The testdb program is executed automatically and exits after it has checked the whole database or the specified table (the `-t` option).

Testdb Messages

Every message produced by the testdb program has its own level of importance. It can be one of the following:

- Fatal error system database is damaged and testing is impossible.
- Serious error These 2 messages indicate that parts of the database structure contain damage that may lead to error generation or incorrect results during kernel execution.
- Error Usually these errors need correcting
- Warning These 2 messages indicate the presence of factors that may lead to lower Linter kernel productivity, but that will not generate errors or incorrect results. These messages may be ignored.
- Information

Below is a listing of testdb messages with short descriptions and recommendations for error correction, where possible.

For errors generated by a specific table, it is advisable to save the table data using the ldb program, then rebuild the table and load it with the saved data.

In this list of messages, the word key refers to options to the testdb command unless otherwise noted.

<u>Message</u>	<u>Reason and Corrective Action</u>
Incorrect number of parameters!	An error in the command line. Launch testdb with the -h option and check the correct syntax.
Incorrect key!	
Key value missing!	
Not enough memory in pool!	The pool size has been set too large or too small. Try to set a new size for the pool using the -p key.
Not enough memory for queues in the pool!	
Cannot open \$\$\$SYSRL. testdb cannot find the DB.	Check the value of SY00 or change to DB's directory and relaunch testdb.
Database blocked by another user.	This occurs when working on a local network.
System database damaged.	Additional messages will be displayed regarding DB element read errors.
Could not read row zero!	These messages may be generated by a variety of circumstances. Usually, they will be followed by additional messages.
Could not extract row zero!	
Could not read SYSREL description!	
Could not extract SYSREL description!	
Could not read SYSATR description!	
Could not extract SYSATR description!	
Could not read SYSUSR description!	

<u>Message</u>	<u>Reason and Corrective Action</u>
Could not extract SYSUSR description!	
Could not read table description!	
Could not read column description!	
System database damaged or its logical integrity compromised.	
Could not open file!	
Specified file missing or blocked.	
Could not read page!	
Illegal number of index files	The table description contains an incorrect number of files.
Illegal number of data files	There may be from 1 to 3 data and index files. There may be from 0 to 3 VAR files. Otherwise, there is an informational error regarding a file.
Illegal number of BLOB files	
File size different in table and file description	
File size larger than in the description	See immediately above.
File length incorrect, true length <length>	
Column has an illegal type	The column description in the \$\$\$ATTRI table contains an illegal Linter type value or length value.
Column has an illegal length	
Column move incorrectly shown in entry	There is an inconsistency between the table description and the descriptions of its columns.
Incorrect entry length	
Incorrect number of indexes	
File contains a non-integer number of pages (bytes)	These messages reflect errors in the bitmap word status and bitmap status. When a bit is set and erroneously cleared, an error is generated. This type of warning may be ignored.
Possibly specified file damaged.	
Bit cleared for a page not in file	
Bit cleared for page not in bitmap	
Bit cleared for full page	
Bit set for a non-full page	
Bit set for a non-full page (bytes)	
Bit cleared for a page contained in the bitmap or the converter	
File size too small for converter.	address converter. In the last two cases, it is advisable to try to use the REBUILD TABLE command.
Converter element contains incorrect	

<u>Message</u>	<u>Reason and Corrective Action</u>
file number.	
Converter element contains incorrect page number.	
Incorrect maximum occupied ROWID value.	This informational message may be ignored.
Incorrect number of occupied ROWID value.	Try the REBUILD TABLE command.
Could not read data!	Inconsistency between the data file and the converter. These may usually be corrected via REBUILD TABLE.
ROWID was not found in data page.	
Entry contains ROWID not specified in the converter.	
ROWID found several times in data page.	Inconsistency between the data file and the converter.
Incorrect page length.	These messages all reflect errors in data file structure.
Entry has illegal length.	
Entry contains an illegal ROWID value.	
Illegal row number.	
Illegal length value.	
Could not read index!	Errors in indexes. Try to delete and rebuild the index.
Index contains an illegal ROWID value.	
Index contains ROWID not present in the converter.	
Index contains an incorrect value.	
Incorrect value order in index.	
ROWID not specified in index.	
ROWID specified several times in index.	
Illegal BLOB value size.	
Incorrect file number.	
Incorrect first page number.	
Incorrect last page number.	
Could not find section with this ROWID.	
Reference to previous page <reference>, was different.	

<u>Message</u>	<u>Reason and Corrective Action</u>
Incorrect reference to next page.	
Incorrect segment size, must be 4048 (page len. – len. of page header).	The word segment in these three messages refers to a slice or chunk of data that is not sized properly. Position refers to the position of an internal structure.
Incorrect last segment size, must be <size>.	
Incorrect segment position (shift, length, page length).	
BLOB length incorrect, length is nnnn	