

Database
Management
System

LINTER[®]

Version 5.9

TCL/TK interface

Relational Expert Systems



Table of Contents

Introduction	3
Return Codes	3
Commands	4
Open Connection.....	4
Close Connection	4
Open Cursor	4
Close Cursor.....	4
Execute Query.....	4
Fetch	5
Get a Row.....	5
Get Single Column Value.....	5
Create Statement	5
Free Statement.....	6
Execute a Statement.....	6
Bind Variable to Query Parameter	6
Get Cursor Option	6
Get a Statement Option.....	6
Commit for Connection.....	7
Commit for Cursor	7
Rollback for Connection	7
Rollback for Cursor.....	7
Free All.....	7

Introduction

LINTCL provides a set of commands to access Linter from Tcl.

Return Codes

All commands return a positive value if they are completed successfully. Otherwise, the code returned will be negative.

The possible values returned on error are:

- 1 LinAPI error.
- 2 Invalid identifier.
- 3 Object is busy.
- 4 Not enough memory.
- 5 LINTCL internal error.

Commands

Open Connection

Syntax

```
linter_connect username ?name length? ?password? \  
?pass length? ?server? ?mode?
```

Opens a connection to Linter server, server, for user, username, and password, password. The mode argument sets one of the four types of transaction processing:

- mOptimistic;
- mExclusive;
- mAutocommit;
- mAnsi.

On successful completion, the positive value returned is the ID of the connection.

Close Connection

Syntax

```
linter_closeconnect ConnectID
```

Closes the ConnectID connection and all cursors and operators associated with it.

Open Cursor

Syntax

```
Linter_opencursor ConnectID ?mode? ?cursor name? ?name length?
```

Opens a cursor for the ConnectID connection in mode `mode`, and with the name `cursor name`. See section “Open Connection” for modes.

The positive value returned on success is the cursor's ID.

Close Cursor

Syntax

```
Linter_closecursor CursorID
```

Closes the cursor having CursorID.

Execute Query

Syntax

```
linter_execdir CursorID query
```

Executes a query for the specified cursor.

Fetch

Syntax

```
linter_fetch CursorID ?direction? ?position?
```

Move to the row in a sample specified by the direction and position argument(s). The values of the direction and position arguments, and their functions, are:

<u>Direction</u>	<u>Position</u>	<u>Function</u>
toNext	space	Move to the next row.
toPrevious	space	Move to the prior row.
toFirst	space	Move to the first row of the sample.
toLast	Space	Move to the last row of the sample.
toAbsNumber	unsigned int	Move to an absolute row number.
toRelNumber	signed int	Move forward (+) or backward (-) the specified number of rows.

Get a Row

Syntax

```
linter_getrow CursorID
```

Returns a list, the first element of which is the completion code. The remainder of this list consists of the current answer row from the cursor CursorID.

Get Single Column Value

Syntax

```
linter_getdata CursorID ColumnNumber ?type?
```

Returns a two-element list, where the first element is the completion code and the second is the value of the column of the answer row.

Type may be either tChar or tString. The default is tChar.

Create Statement

Syntax

```
linter_createstmt ConnectID query
```

Translates an SQL query and creates a statement. Returns the ID of the statement.

Free Statement

Syntax

```
linter_freestmt StmtID
```

Frees the statement specified by StmtID.

Execute a Statement

Syntax

```
linter_execstmt CursorID StmtID
```

Executes the statement StmtID on the cursor CursorID.

Bind Variable to Query Parameter

Syntax

```
linter_bindpar CursorID StmtID number var
```

Binds the variable var to the query parameter numbered number.

Get Cursor Option

Syntax

```
linter_cursopt CursorID option ?number?
```

Returns a list in which the first element is the completion code and the second is the value of the required cursor option. If the option type is cAnswerDesc, it is necessary to indicate the number of the answer column. The value will be stored in seven elements of the list, each representing one field of t_ParamDesc structure.

The types of options are similar to Linter API options. See document “Linter API”, section “Statements” for the t_ParamDesc structure.

Get a Statement Option

Syntax

```
linter_stmtopt StmtID option ?number?
```

Returns a list in which the first element is the completion code and the second is the value of the required option. If the option type is cAnswerDesc, the number of the answer column must be indicated. The values will be stored in five elements of the list corresponding to the fields Owner, Table, Column, Type and Length of the t_ParamDesc structure.

If the option type is sParamDesc, the parameter number must be indicated. The value will be stored in three elements of the list corresponding to the Column, Type and Length fields of the t_ParamDesc structure.

The types of options are similar to Linter API options. See document “Linter API”, section “Statements” for the `t_ParamDesc` structure.

Commit for Connection

Syntax

```
linter_commit ConnectID
```

Commits transaction results for all cursors on the specified connection.

Commit for Cursor

Syntax

```
linter_commcurs CursorID
```

Commits the results of the transaction on the cursor `CursorID`.

Rollback for Connection

Syntax

```
linter_rollback ConnectID
```

Rolls back transaction results for all cursors on the connection `ConnectID`.

Rollback for Cursor

Syntax

```
Linter_rollbcurs CursorID
```

Rolls back the results of a transaction on cursor `CursorID`.

Free All

Syntax

```
linter_freeall
```

Closes all connections, cursors, and operators. Frees all resources currently in use. This command must be issued before quitting the application.