

**МОБИЛЬНАЯ
РЕЛЯЦИОННАЯ
СУБД**

ЛИНТЕР®

Linter Standard
Linter Bastion
Linter RealTime
Linter Multiversion

TCL/TK-интерфейс

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

 **РЕЛАКС®**

Товарные знаки

РЕЛЭКС™, ЛИНТЕР® , НЕВОД® , LAV™, ЛАКУНА являются товарными знаками, принадлежащими ЗАО НПП «Реляционные экспертные системы» (далее по тексту – компания РЕЛЭКС). Прочие названия и обозначения продуктов являются товарными знаками их производителей, продавцов или разработчиков.

Интеллектуальная собственность

Правообладателем продуктов ЛИНТЕР®, НЕВОД®, LAV™, ЛАКУНА является компания РЕЛЭКС (1990–2011). Все права защищены. Данный документ является собственностью компании РЕЛЭКС. Ни одна часть данного документа не может быть воспроизведена, передана, преобразована, сохранена в системе поиска информации, переведена на другой язык или компьютерный язык в какой-либо форме, какими-либо средствами, электронными, механическими, магнитными, оптическими, химическими, ручными или иными без предварительного разрешения компании РЕЛЭКС.

О документе

Материал, содержащийся в данном документе, прошел тщательную проверку, но компания РЕЛЭКС не гарантирует, что документ не содержит ошибок и пропусков. Компания РЕЛЭКС оставляет за собой право в любое время вносить в документ исправления и изменения, пересматривать и обновлять содержащуюся в нем информацию.

Адрес

394006, г. Воронеж, ул. 20-летия Октября, 119.

Тел./факс: (473) 2-711-711, 2-778-333.

e-mail: market@relex.ru.

Адрес для корреспонденции

394000, г. Воронеж, а/я 137.

Техническая поддержка

Отдел поддержки и сопровождения программных продуктов:

телефон: (473) 2-711-711 с 9:00 до 18:00 мск.

e-mail: support@relex.ru, market@relex.ru.

С целью повышения качества разрабатываемых программных средств и предоставляемых услуг в компании РЕЛЭКС действует автоматизированная система учёта и обработки рекламаций. Обо всех обнаруженных недостатках и ошибках в программном продукте и/или документации на него просим сообщать нам на Internet-странице [рекламация](#).

Оглавление

Предисловие	1
Назначение документа.....	1
Для кого предназначен документ.....	1
Необходимые предварительные знания.....	1
Принятые обозначения и соглашения.....	1
Дополнительные документы.....	2
Общие сведения	3
Сборка приложения	4
Функции	5
linter_connect.....	5
Linter_opencursor.....	6
Linter_closecursor.....	6
Linter_execdir.....	7
Linter_fetch.....	7
Linter_getrow.....	8
Linter_getdata.....	8
Linter_createstmt.....	9
Linter_freestmt.....	9
Linter_execstmt.....	10
Linter_bindpar.....	10
Linter_cursopt.....	11
Linter_stmtopt.....	13
Linter_commit.....	14
Linter_commcurs.....	15
Linter_rollback.....	15
Linter_rollbcurs.....	16
Linter_freeall.....	16
Обработка кодов завершения	17

Предисловие

Назначение документа

Документ содержит описание TCL/TK–интерфейса с СУБД ЛИНТЕР.

Документ может использоваться для работы с любой версией СУБД ЛИНТЕР. Особенности конкретных версий оговариваются по тексту.

Для кого предназначен документ

Документ предназначен для программистов, разрабатывающих приложения на языке программирования TCL/TK с использованием СУБД ЛИНТЕР.

Необходимые предварительные знания



Для работы с интерфейсом необходимо знать;

- основы реляционных баз данных и языка баз данных SQL;
- язык программирования TCL/TK;
- уметь работать в соответствующей операционной системе на уровне простого пользователя.

Принятые обозначения и соглашения

<u>Обозначение</u>	<u>Пример</u>	<u>Значение</u>
Курсив	Растровым называется изображение...	Новый термин в тексте
Полужирный шрифт	В этом случае необходимо переносить все физические файлы.	Выделение в тексте
Подчеркнутый шрифт	Подробную информацию о работе программы можно получить на сайте www.dmk.ru .	Адреса страниц Internet
Текст, разделенный знаком ⇒	Выполните команду View ⇒ Properties (Вид ⇒ Свойства).	Последовательность выполнения команд
Текст, заключенный в <>, со знаком + между ними	<Ctrl>+<C>	В <> заключаются клавиши клавиатуры, знак + означает сочетание клавиш
Крупный моноширинный текст	SQL> _q	Текст командной строки
Мелкий моноширинный текст	Page Time Count	Текст программы

Предисловие

Обозначение	Пример	Значение
Заглавные буквы	BROWSE	Названия команд, слова, зарезервированные в SQL, ключевые слова
Курсив в <>	<return statement>	Определяемый элемент синтаксической конструкции
Символ ::=		Равенство по определению. Слева от знака стоит определяемое понятие, справа – собственно определение понятия
Квадратные скобки []	DBSTORE [-d -n -o -p -r -t -u]	Необязательные элементы конструкции. В данном примере ключи не являются обязательными элементами команды
Вертикальная черта 	<return value> ::= <value expression> NULL	Указывает на то, что все предшествующие ей элементы списка являются необязательными и могут быть заменены любым другим элементом списка после этой черты
Фигурные скобки { }	CODEPAGE { 866 1251 KOI8}	Указывают на то, что все находящееся внутри них является единым целым
Многоточие «...»	Характеристики столбца MAKE CHAR(20) MODEL CHAR(20) ... SQL>	Означает, что предшествующая часть может быть повторена любое количество раз
Многоточие, внутри которого находится запятая «,...»		Указывает, что предшествующая часть оператора, состоящая из нескольких элементов, разделенных запятыми, может иметь произвольное число повторений
Текст со знаком  на сером фоне	 Если конфигурация страницы-шаблона не учитывала свойств, команда будет выполнена некорректно.	Примечание

Дополнительные документы

- СУБД ЛИНТЕР. Архитектура СУБД.
- СУБД ЛИНТЕР. Справочник по SQL.
- СУБД ЛИНТЕР. Справочник кодов завершения.

Общие сведения

Интерфейс LinTCL представляет собой набор команд, обеспечивающих доступ из программы на языке программирования TCL к СУБД ЛИНТЕР.

Все команды в случае удачного завершения возвращают положительное значение, в противном случае код возврата будет отрицательным.

Возможные возвращаемые значения в случае ошибки:

- -1 – ошибка LinAPI;
- -2 – неверный идентификатор;
- -3 – объект занят;
- -4 – нет памяти;
- -5 – внутренняя ошибка LinTCL.

Сборка приложения

Для сборки приложения на языке TCL с поддержкой доступа к СУБД ЛИНТЕР:

- 1) выполнить файл `configure` из корневого каталога дистрибутива (если он не выполнялся ранее);
- 2) перейти в каталог `tcl`;
- 3) выполнить команду `make`, предварительно проверив наличие в каталоге `bin` дистрибутива ЛИНТЕР файлов:

`linsh` - TCL интерпретатор;

`linwish` - ТК интерпретатор с поддержкой СУБД ЛИНТЕР

Если операционная система поддерживает разделяемые библиотеки, то возможен доступ к СУБД ЛИНТЕР в виде пакета (`package`) TCL.

Для этого:

- 1) выполнить файл `configure` из корневого каталога дистрибутива (если он не выполнялся ранее);
- 4) перейти в каталог `tcl`;
- 5) войти в ОС под привилегированным пользователем и выполнить команду `make install`

В результате в каталоге пакетов TCL (зависит от ОС) будет создан подкаталог `tclLinter` с двумя файлами `libtclLinter.so` и `pkgIndex.tcl`.

- 6) перед выполнением любой команды доступа к СУБД ЛИНТЕР в TCL-скрипт включить строку `"package require tclLinter"`



Предполагается, что TCL/ТК уже установлены в вашей системе.

Функции

linter_connect

Назначение

Установить соединение с БД.

Синтаксис

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

Параметр	Описание
<code>username</code>	Имя пользователя БД (регистр символов учитывается) длиной не более 66 символов. Если длина имени больше 66 символов, то она усекается.
<code>name length</code>	Длина параметра <code>username</code> . Если задано 0—берется фактическая длина <code>username</code> , но не более 66 символов. Если фактическая длина <code>username</code> больше <code>name length</code> , то <code>username</code> усекается, если меньше – дополняется пробелами справа.
<code>password</code>	Пароль пользователя БД (регистр символов учитывается) длиной не более 18 символов. Если длина пароля больше 18 символов, то она усекается.
<code>pass length</code>	Длина параметра <code>password</code> . Если задано 0—берется фактическая длина <code>password</code> , но не более 18 символов. Если фактическая длина <code>password</code> больше <code>pass length</code> , то <code>password</code> усекается, если меньше – дополняется пробелами справа.
<code>server</code>	Имя БД длиной не более 8 символов или " для соединения по умолчанию. Если длина имени больше 8 символов, то она усекается.
<code>mode</code>	Режим обработки транзакций: <ul style="list-style-type: none">• <code>mOptimistic</code>• <code>mExclusive</code>• <code>mAutocommit</code>• <code>mAnsi</code>. Значение по умолчанию <code>mAutocommit</code> .

Функции

Возвращаемые значения

Значение	Описание
>0	Идентификатор соединения (нормальное завершение).
<0	Код завершения (ошибка выполнения команды).

Lintor_opencursor

Назначение

Открыть курсор по указанному соединению.

Синтаксис

```
lintor_opencursor ConnectID ?mode? ?cursor name? ?name length?
```

Параметр	Описание
ConnectID	Идентификатор соединения.
mode	Режим обработки транзакций в курсоре: <ul style="list-style-type: none">• mOptimistic• mExclusive• mAutocommit• mAnsi. Значение по умолчанию mAutocommit.
cursor name	Имя курсора(регистр символов учитывается) длиной не более 66 символов. Если длина имени курсора больше 18 символов, то она усекается.
name_length	Длина параметра cursor name. Если задано 0–берется фактическая длина cursor name , но не более 66 символов. Если фактическая длина cursor name больше name length, то cursor name усекается, если меньше – дополняется пробелами справа.

Возвращаемые значения

Значение	Описание
>0	Идентификатор курсора (нормальное завершение).
<0	Код завершения (ошибка выполнения команды).

Lintor_closecursor

Назначение

Закрывает указанный курсор.

Синтаксис

```
lintor_closecursor CursorID
```

Параметр	Описание
<code>CursorID</code>	Идентификатор открытого курсора.

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Lintor_execdir

Назначение

Выполнить без предварительной подготовки SQL-запрос по указанному курсору.

Синтаксис

```
lintor_execdir CursorID query
```

Параметр	Описание
<code>CursorID</code>	Идентификатор открытого курсора.
<code>query</code>	Текст SQL-оператора (должен заканчиваться ;).

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Lintor_fetch

Назначение

Перейти в заданную строку ответа курсорного запроса.

Синтаксис

```
lintor_fetch CursorID ?direction? ?position?
```

Параметр	Описание
<code>CursorID</code>	Идентификатор открытого курсора.
<code>direction</code>	Направление перемещения в выборке: <ul style="list-style-type: none"> • <code>toNext</code> - на следующую строку после текущей ; • <code>toPrevious</code> - на предыдущую после текущей; • <code>toFirst</code> - на первую ; • <code>toLast</code> - на последнюю; • <code>toAbsNumber</code> - на строку с абсолютным номером

Функции

Параметр	Описание
<code>position</code>	position; <ul style="list-style-type: none">• <code>toRelNumber</code> - на строку с номером <code>position</code> относительно текущей строки. Номер строки (для параметров <code>toAbsNumber</code> , <code>toRelNumber</code>). Нумерация начинается с 1.

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Linter_getrow

Назначение

Получить текущую запись курсорного запроса.

Синтаксис

```
linter_getrow CursorID
```

Параметр	Описание
<code>CursorID</code>	Идентификатор курсора, по которому был выполнен SQL-запрос выборки данных (SELECT).

Возвращаемые значения

Массив значений, первый элемент которого - код завершения команды (0 – нормальное завершение, <0 – ошибка выполнения команды), а остальные элементы массива – значения столбцов текущей записи выборки.

Linter_getdata

Назначение

Получить значение заданного столбца текущей записи курсорного запроса.

Синтаксис

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

Параметр	Описание
<code>CursorID</code>	Идентификатор курсора, по которому был выполнен SQL-запрос выборки данных (SELECT).
<code>ColumnNumber</code>	Номер столбца. Нумерация начинается с 1.
<code>type</code>	Тип представления возвращаемого значения: <ul style="list-style-type: none">• <code>tChar</code>;• <code>tString</code>.

Возвращаемые значения

Массив значений из двух элементов.

Первый элемент – код завершения команды (0 – нормальное завершение, <0 – ошибка выполнения команды), второй – значение указанного столбца текущей записи выборки.

Lintestmt

Назначение

Подготовить SQL-оператор к выполнению.

Синтаксис

```
lintestmt ConnectID query
```

Параметр	Описание
ConnectID	Идентификатор соединения, по которому выполняется подготовка SQL-оператора.
query	Текст SQL-оператора.

Возвращаемые значения

Значение	Описание
>0	Идентификатор оператора (нормальное завершение).
<0	Код завершения (ошибка выполнения команды).

Lintestmt

Назначение

Освободить подготовленный SQL-оператор.

Синтаксис

```
lintestmt StmtID
```

Параметр	Описание
StmtID	Идентификатор подготовленного SQL-оператора.

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Linteexecstmt

Назначение

Выполнить подготовленный SQL-оператор.

Синтаксис

```
linter_execstmt CursorID StmtID
```

Параметр	Описание
CursorID	Идентификатор открытого курсора, по которому должен быть выполнен SQL-оператор.
StmtID	Идентификатор подготовленного SQL-оператора..

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Lintebindpar

Назначение

Привязка переменной к параметру SQL-оператора.

Синтаксис

```
linter_bindpar CursorID StmtID number var
```

Параметр	Описание
CursorID	Идентификатор открытого курсора, к которому выполняется привязка.
StmtID	Идентификатор подготовленного SQL-оператора.
number	Номер привязываемого параметра. Нумерация параметров начинается с 1.
var	Имя переменной, привязываемой к параметру.

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Lintер_cursoropt

Назначение

Получить характеристики курсора.

Синтаксис

```
lintер_cursoropt CursorID option ?number?
```

Параметр	Описание
CursorID	Идентификатор курсора.
option	Идентификатор характеристики курсора.
number	Номер столбца курсора (только для характеристики cAnswerDesc). Нумерация столбцов с 1.

Идентификатор характеристики	Описание
cSelectRowCount	Число строк в выборке.
cRowCount	Количество обработанных последним SQL-оператором записей.
cColumnCount	Количество столбцов в ответе.
cStmtType	Тип последнего SQL-оператора.
cCurrentRow	Номер (в выборке) текущей записи.
cCurrentRowID	RowId текущей записи в таблице (для однопеременных Select-запросов).
cExecStmt	Количество выполненных (за один Execute) операторов.
cExecRow	Количество записей обработанных (за один Execute).
cAnswerDesc	Описание столбца ответа.
cAnswerSize	Размер записи ответа (в байтах).
cStmtNumber	Номер текущего оператора.
cParamNumber	Номер параметра, при обработке которого произошла ошибка.
cPriority	Приоритет курсора.
cConnectID	Номер соединения, по которому открыт курсор.
cTransMode	Режим обработки транзакций курсора.
cApiCode	Код завершения LinAPI.
cLinCode	Код завершения СУБД ЛИНТЕР при последней операции по курсору.
cSysCode	Код сообщения ОС при последней операции по курсору.
cStrNumber	Номер строки запроса, где встречена ошибка

<u>Идентификатор характеристики</u>	<u>Описание</u>
cPosNumber	трансляции. Номер позиции в строке запроса, где встречена ошибка трансляции.
cCursorName	Имя курсора.
cNullIndictor	Признак NULL-значения.
cIsAutoInc	Признак AUTOINC-значения.
cProcErrNum	Количество ошибок при трансляции процедуры.
cProcArgNum	Количество аргументов в процедуре.
cProcArgDesc	Описание параметра процедуры.
cProcArgName	Имя параметра процедуры.
cProcCursorArg	Имя курсора процедуры.
cWaitComplete	Ждать завершения асинхронной операции.
cWaitTimeout	Длительность тайм-аута.
cUserData	Пользовательские данные.
cIsUpdatable	Допустимость корректировки данных.
cBlobLength	Длина BLOB-данных.
cBlobType	Тип BLOB-данных.

Возвращаемые значения

Массив значений из двух элементов.

Первый элемент – код завершения команды (0 – нормальное завершение, <0 – ошибка выполнения команды), второй – значение требуемой характеристики курсора.

Если тип характеристики cAnswerDesc, то значение возвращается в виде структуры:

```
typedef struct {
    L_CHAR      Owner[66];
    L_CHAR      Table[66];
    L_CHAR      Column[66];
    L_WORD      Length;
    L_BYTE      Type;
    L_BYTE      Prec;
    L_BYTE      Scale;
    L_BYTE      NullIndicator;
    L_LONG      RealLength;
} t_ParamDesc;
```

Если тип характеристики cProcArgDesc, то значение возвращается в виде структуры:

```
typedef struct
{
    BYTE    Flags, /* флаги значения */
    BYTE    Reserv, /* зарезервировано*/
    WORD    Value, /* относительный адрес значение в */
                /* буфере ответа */
    t_ArgType Type, /* описание типа значения */
} t_ProcArgDesc;
```

Флаги значения

fNULL 0x02 /* NULL-значение */

fCursor 0x04 /* значение является курсором */

fName 0x08 /* есть информация об имени выходного параметра */

Характеристика параметра процедуры

```
BYTE    Ntype, /* тип значения */
        Prec, /* точность */
        Scale, /* масштаб */
        Reserv; /* зарезервировано */
} t_ArgType;
```

Lintert_stmtopt

Назначение

Получить характеристики SQL-оператора.

Синтаксис

```
lintert_stmtopt StmtID option ?number?
```

Параметр	Описание
StmtID	Идентификатор предложения.
option	Идентификатор характеристики предложения.
number	Номер столбца ответа предложения (только для характеристики cAnswerDesc). Нумерация столбцов с 1.

Идентификатор характеристики

Описание

sParamCount	Число параметров в операторе.
sAnswerDesc	Описание ответа.
sColumnCount	Число столбцов в ответе.
sParamDesc	Описание параметра.
sConnectID	№ соединения, по которому создан оператор.
sApiCode	Код сообщения LinAPI.
sIsAutoInc	Признак AUTOINC-значения.
sStmtType	Тип оператора.
sParamType	Тип параметра (pInput, pOutput, pInputOutput).
sUserData	Пользовательские данные.

Возвращаемые значения

Массив значений из двух элементов.

Первый элемент – код завершения команды (0 – нормальное завершение, <0 – ошибка выполнения команды), второй – значение требуемой характеристики курсора.

Если тип характеристики sAnswerDesc, то значение возвращается в виде структуры:

```
typedef struct {
    L_CHAR Owner[66];
    L_CHAR Table[66];
    L_CHAR Column[66];
    L_BYTE Type;
    L_WORD Length;
};
```

Если тип характеристики sParamDesc, то значение возвращается в виде структуры:

```
typedef struct {
    L_CHAR Column[66];
    L_BYTE Type;
    L_WORD Length;
};
```

Lint_commit

Назначение

Зафиксировать в БД результаты транзакций всех курсоров данного соединения.

Синтаксис

```
lint_commit ConnectID
```

Параметр

Описание

ConnectID

Идентификатор соединения.

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Lintер_commcurs

Назначение

Зафиксировать в БД результаты транзакции заданного курсора.

Синтаксис

```
linter_commcurs CursorID
```

Параметр	Описание
CursorID	Идентификатор курсора.

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Lintер_rollback

Назначение

Отменить результаты транзакций всех курсоров данного соединения.

Синтаксис

```
linter_rollback ConnectID
```

Параметр	Описание
ConnectID	Идентификатор соединения.

Возвращаемые значения

Значение	Описание
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Linter_rollbcurs

Назначение

Отменить результаты транзакции заданного курсора.

Синтаксис

```
linter_rollbcurs CursorID
```

<u>Параметр</u>	<u>Описание</u>
CursorID	Идентификатор курсора.

Возвращаемые значения

<u>Значение</u>	<u>Описание</u>
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Linter_freeall

Назначение

Закрыть все соединения, курсоры, операторы и освободить все занятые при работе ресурсы. Эту команду необходимо подавать перед завершением работы приложения.

Синтаксис

```
linter_freeall
```

Возвращаемые значения

<u>Значение</u>	<u>Описание</u>
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

Обработка кодов завершения

Все команды TCL-интерфейса в случае возникновения ошибочной ситуации при работе с базой данных возвращают отрицательный результат, но интерпретатор TCL не воспринимает это как ошибку и программисту необходимо самому анализировать и обрабатывать такой код завершения. Для этого предназначена команда `linter_error`.

Она имеет две функции:

- 1) расшифровка кода завершения. В случае если при выполнении предыдущей команды семейства `linter_...` возникла ошибка при работе с БД, то интерпретатору также сообщается об ошибочной ситуации и передается сообщение с расшифровкой ошибки;
- 2) фильтрация результатов ответа для поиска определенной информации.

В этом случае в качестве параметров команде `linter_error` можно передать результат выполнения одной из команд `linter_...`, и в случае успешного завершения команды, `linter_error` выполняет роль фильтра, вырезающего из результата код завершения и возвращающего только полезную информацию. Например, узнать количество ответов на запрос можно следующим образом

```
set countAnswer [linter_error [linter_cursopt $cursor  
cSelectRowCount]]
```

Назначение

Получить результат выполнения одной из предыдущих команд TCL-интерфейса.

Синтаксис

```
linter_error parameters
```

Возвращаемые значения

<u>Значение</u>	<u>Описание</u>
0	Нормальное завершение.
<0	Код завершения (ошибка выполнения команды).

