СИСТЕМА
УПРАВЛЕНИЯ
БАЗАМИ
ДАННЫХ

ЛИНТЕР®

ЛИНТЕР БАСТИОН

ЛИНТЕР СТАНДАРТ

Создание и конфигурирование базы данных

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

РЕЛЭКС

Товарные знаки

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

Интеллектуальная собственность

Правообладателем продуктов ЛИНТЕР® является компания РЕЛЭКС (1990-2026). Все права защищены.

Данный документ является результатом интеллектуальной деятельности, права на который принадлежат компании РЕЛЭКС.

Все материалы данного документа, а также его части/разделы могут свободно размещаться на любых сетевых ресурсах при условии указания на них источника документа и активных ссылок на сайты компании РЕЛЭКС: relex.ru и linter.ru.

При использовании любого материала из данного документа несетевым/печатным изданием обязательно указание в этом издании источника материала и ссылок на сайты компании РЕЛЭКС: relex.ru и linter.ru.

Цитирование информации из данного документа в средствах массовой информации допускается при обязательном упоминании первоисточника информации и компании РЕЛЭКС.

Любое использование в коммерческих целях информации из данного документа, включая (но не ограничиваясь этим) воспроизведение, передачу, преобразование, сохранение в системе поиска информации, перевод на другой (в том числе компьютерный) язык в какой-либо форме, какими-либо средствами, электронными, механическими, магнитными, оптическими, химическими, ручными или иными, запрещено без предварительного письменного разрешения компании РЕЛЭКС.

О документе

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

Контактные данные

394006, Россия, г. Воронеж, ул. Бахметьева, 2Б.

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

e-mail: info@linter.ru.

Техническая поддержка

С целью повышения качества программного продукта ЛИНТЕР и предоставляемых услуг в компании РЕЛЭКС действует автоматизированная система учёта и обработки пользовательских рекламаций. Обо всех обнаруженных недостатках и ошибках в программном продукте и/или документации на него просим сообщать нам в раздел Поддержка на сайте ЛИНТЕР.

Содержание

Предисловие 2
Назначение документа 2
Для кого предназначен документ 2
Необходимые предварительные знания 2
Условия выполнения 3
Выполнение программы 4
Запуск 4
Команды 5
Общие сведения 5
Пароль защищенной БД 5
Путь к БД 6
Создание БД 7
Конфигурирование БД 13
Установка параметров конфигурирования 14
Просмотр параметров конфигурирования 20
Справочная информация 23
Синтаксис команд 23
Переменные среды окружения 24
Завершение работы 25
Ключи 25
Полная информация о версии СУБД ЛИНТЕР 25
Краткая информация о версии СУБД ЛИНТЕР 25
Кодовая страница 26
Значение параметра конфигурации 26
Сообщения и коды завершения 28
Информационные сообщения 28
Сообщения об ошибках 28
Коды завершения 29
Приложение 1. Список ключевых слов и их сокращений 30
Приложение 2. Алгоритм определения местоположения БД 31

Предисловие

Назначение документа

Документ содержит описание утилиты `gendb`, выполняющей создание системных таблиц и рабочих файлов базы данных (БД) ЛИНТЕР и конфигурирование созданной БД для оптимизации работы с ней.

Документ предназначен для СУБД ЛИНТЕР БАСТИОН 6.0 сборка 20.8, далее по тексту СУБД ЛИНТЕР.

Для кого предназначен документ

Документ предназначен для системных администраторов СУБД ЛИНТЕР.

Необходимые предварительные знания

Для работы необходимо:

- знать основы реляционных БД;
- знать и понимать архитектуру СУБД ЛИНТЕР (для её тонкой настройки);
- уметь работать в соответствующей операционной системе на уровне простого пользователя.

Дополнительные документы

- Тестирование базы данных
- Сетевой администратор
- Справочник по SQL
- Администрирование комплекса средств защиты данных
- Рекомендации по настройке СУБД ЛИНТЕР

Условия выполнения

Утилита должна запускаться при неактивном ядре СУБД ЛИНТЕР.

Выполнение программы

Запуск

Запуск утилиты осуществляется стандартными средствами запуска задач, имеющимися в каждой операционной системе.

Исполняемый файл утилиты – `gendb`.

Возможны следующие способы запуска утилиты:

1) пакетный режим:

```
gendb <спецификация файла>
```

Утилита выполняет команды указанного файла и возвращает управление операционной системе.

2) командный режим:

```
gendb <ключ>
```

Утилита обрабатывает заданный ключ и возвращает управление операционной системе.

Ключи используются для передачи утилите коротких инструкций (см. раздел Ключи).

3) интерактивный режим:

```
gendb <Enter>
```

Утилита переходит в режим интерактивного ввода команд (выдает приглашение):

```
gendb>
```

При ожидании утилитой продолжения ввода команды на экран выводится подсказка:

```
_gendb>
```

Команды используются для передачи утилите инструкций, текст которых может достигать нескольких экранных строк (см. раздел Команды).

В ОС Windows файлы, расположенные в системных каталогах `%systemroot%`, `%ProgramFiles%`, защищены средствами ОС от изменения. Для внесения изменений в защищаемые файлы приложение должно запускаться от имени администратора ОС, в противном случае изменяемые данные будут размещаться в отдельном виртуализированном представлении ресурса, являющемся копией изменяемого ресурса. Виртуализированная копия поддерживается для профиля пользователя. В итоге для каждого пользователя, выполняющего несовместимое приложение, создается отдельная копия виртуализированного файла.

Применительно к утилите `gendb`, запущенной от имени простого пользователя, это означает, что изменения, вносимые в БД, размещенную в защищаемой папке (например, `c:\program files\linter\db`) при её настройке (конфигурировании), будут храниться не в самой БД, а в её отдельной виртуализированной копии. То есть в данном случае внесенные утилитой `gendb` изменения СУБД ЛИНТЕР видеть не будет.

Команды

Общие сведения

Ключевые слова команды допускается вводить как малыми, так и большими буквами, при этом верхний и нижний регистр не различаются.

Значения параметров команды регистрозависимы, поэтому, при необходимости, они должны заключаться в двойные кавычки.

После ввода команды синтаксический анализатор утилиты производит разбор выражения путём поиска ключевых слов, которые можно задавать в сокращенном виде. Применять сокращения можно в том случае, если по первым введённым символам синтаксический анализатор утилиты сможет однозначно идентифицировать ключевое слово.

Список ключевых слов команд утилиты и их максимально возможные сокращения приведены в приложении 1.

Все элементы команд, которые не распознаны как ключевые слова, воспринимаются в виде строковых констант. В том случае, если строковая константа содержит специальные символы, она должна быть заключена в двойные кавычки.

Команда включает имя команды и параметры. Текст команды может занимать несколько строк и должен обязательно заканчиваться знаком «точка с запятой» (;), являющимся признаком конца команды.

Команды `HELP` и `EXIT` могут употребляться без знака «точка с запятой» (;).

Примеры

1. Пример команды с полным написанием ключевых слов.

```
gendb>
_gendb> create database "Order Records"
_gendb>maxtab 70 maxcol 500 maxusr 30
_gendb>message output "d:\linter\db\protocol.txt";
gendb>exit
```

2. Пример команды с максимальным сокращением ключевых слов.

```
gendb>
_gendb> cr da "Order Records"
_gendb>maxt 70 maxc 500 maxu 30
_gendb>me ou "d:\linter\db\protocol.txt";
gendb>exit
```

Пароль защищенной БД

Синтаксис

```
<пароль защиты БД>::=
SET PROTECTED DATABASE PASSWORD "<пароль>";
```

```
<пароль>::= [$<алгоритм преобразования>$]<ключ>
<алгоритм преобразования>::= {DES | GOST | AES}
<ключ>::= символьная строка
```

Задает регистрозависимый пароль защищенной БД. Пароль используется в качестве ключа для кодирования/декодирования содержимого страниц файла данных таблиц БД по заданному алгоритму.

Максимальная длина <ключа> 18 символов. Ключ большей длины будет усечен до этого значения.

Если <алгоритм преобразования> не задан, по умолчанию используется `GOST`.

<Пароль> должен задаваться при запуске СУБД на защищенной БД и при запуске утилит, которые работают с БД напрямую, при неактивном ядре СУБД (например, утилита `testdb` (см. документ «Тестирование базы данных», пункт «Тестирование защищенной БД»)).

Команда используется:

1) при создании новой защищенной БД. Защите подвергаются страницы файлов данных таблиц БД;
2) при конфигурировании существующей защищенной БД;
3) при получении информации о параметрах конфигурирования БД.

Примеры

```
gendb>SET PROTECTED DATABASE PASSWORD SALE;
gendb>SET PROTECTED DATABASE PASSWORD "Бухгалтерия";
gendb>SET PROTECTED DATABASE PASSWORD "$DES$56Ew!C+";
```

Путь к БД

Назначение

Для указания местоположения создаваемой (конфигурируемой) БД можно использовать следующие способы:

1) использовать конструкцию <путь к БД>, которая задает местоположение создаваемой или конфигурируемой БД. Заданный путь остается неизменным с момента выполнения команды в течение всего сеанса работы утилиты или до установки нового <пути к БД>. Установленный путь к БД используется всеми остальными командами утилиты, поэтому является глобальным определением местоположения БД;
2) использовать для указания местоположения БД переменную окружения `SY00`, которая в СУБД ЛИНТЕР является переменной окружения по умолчанию. В этом случае `SY00` должна быть предварительно установлена средствами операционной системы;
3) использовать локальное задание пути к БД, которое не отменяет глобальное и действует только в пределах текущей команды утилиты (см. параметр DATABASE DEVICE, параметр ON в команде `CREATE`);
4) если не использован ни один из вышеперечисленных способов задания местоположения БД, то будет использован текущий каталог ОС.

Синтаксис

```
<путь к БД>::=
SET
{DATABASE DIRECTORY "<спецификация каталога>"
| DATABASE PATH <переменная окружения>
| DATABASE DEVICE <переменная БД>};

DATABASE DIRECTORY "<спецификация каталога>"
```

Задает устройство/каталог, на/в котором располагается системная БД, указывая полный путь к этому устройству.

Команда `set database directory` отменяет значение переменной окружения `SY00`. Например, если установлена переменная среды окружения `SY00`, задающая местоположение БД, то после выполнения команды `set database directory` при выполнении нижеперечисленных команд утилиты будет использоваться путь, установленный командой `set database directory`, а не переменной `SY00`:

```
get
set ... device ...
set ... size ...
set syssrt count ...
set logfile ...
DATABASE PATH <переменная окружения>
```

Задает переменную среды окружения, которая определяет устройство/каталог, на/в котором располагается системная БД. Указанная <переменная окружения> предварительно должна быть установлена средствами операционной системы.

```
DATABASE DEVICE <переменная БД>
```

Специфицирует устройство/каталог, на/в котором располагается конфигурируемая системная БД. Указанная спецификация действительна только для тех параметров, которые заданы вместе с ней именно в конкретной командной строке.

Например, `set database device SY01 database name "NEW NAME BD"` меняет имя только той БД, на которую ссылается переменная `SY01`. Для следующей команды `SET` это назначение уже недействительно.

Примечание
В связи с тем, что для представления текста в кириллице в разных операционных системах могут использоваться различные кодировки, рекомендуется для задания пути к БД использовать только латиницу.

Создание БД

Примечание
Создать БД с меньшей детализацией параметров можно с помощью утилиты `linadm` (см. документ «Сетевой администратор», раздел «Создание БД»).

Синтаксис

```
<создание БД>::=
```

```
[<пароль защиты БД>]
[<путь к БД>]
CREATE DATABASE ["<имя БД>"]
[USERNAME "<имя создателя БД>"]
[PASSWORD "<пароль создателя БД>"]
[MESSAGE OUTPUT "<спецификация файла сообщений>"]
[NO MESSAGE]
[NO QUERY]
[ON <переменная окружения>]
[MAXTAB <количество таблиц>]
[MAXCOL <количество столбцов>]
[MAXUSR <количество пользователей>]
[MAXROWSIZE  <максимальный размер записи>]
[CERTIFIED | ORDINARY]
;
```

Каталог, в котором будет создаваться БД, зависит от ряда условий (наличие переменной окружения `SY00`, использование глобальной или локальной установки для указания местоположения БД). Алгоритм выбора каталога создаваемой БД приведен в приложении 2 (см. также команду SET данной утилиты). О выбранном местоположении создаваемой БД на консоль выдается информационное сообщение вида:

Путь к БД 'd:\Program Files\Softool2003608rust\db1'. (переменная окружения `SY01`)

`<пароль защиты БД>`

См. пункт Пароль защищенной БД.

`<путь к БД>`

См. пункт Путь к БД.

`DATABASE "<имя БД>"`

Имя создаваемой БД (не более 18-ти символов). По умолчанию заполняется пробелами. Если строка содержит более 18-ти символов, она усекается до допустимой длины без предупреждения.

В качестве "<имя БД>" допустимо использовать символы из таблицы ASCII.

`USERNAME "<имя создателя БД>"`

<Имя создателя БД> – регистрозависимая строка, содержащая не более 66-ти символов. По умолчанию используется имя `SYSTEM`.

Примеры

```
USERNAME ADMIN
USERNAME "Admin"
USERNAME "Создатель БД"
```

`PASSWORD "<пароль создателя БД>"`

<Пароль создателя БД> – регистрозависимая строка, содержащая не более 18-ти символов. По умолчанию создатель БД имеет пароль `MANAGER8`.

Примеры

```
PASSWORD 12345
PASSWORD 15.07.1995
PASSWORD "67F223$$щгл"
```

```
MESSAGE OUTPUT "<спецификация файла сообщений>"
```

Местоположение выходного файла для записи протокола информационных сообщений, выдаваемых утилитой в процессе работы. Если ключ не задан, информационные сообщения будут выводиться на экран терминала.

```
NO MESSAGE
```

Подавляет выдачу информационных сообщений.

```
NO QUERY
```

Подавляет вопросы на подтверждение создания БД.

```
ON <переменная окружения>
```

Задает имя <переменной окружения>, указывающей местоположение (логическое имя устройства/каталога), где должны создаваться файлы БД (см. приложение 1). В случае, когда модификатор `ON` задаёт устройство, для которого не определена переменная среды окружения и это устройство не совпадает `SY00`, то выдаётся предупреждение.

Например:

```
SET DATABASE DIRECTORY "D:\linter\db\TEST_DB\";
CREATE DATABASE "Database Name" ON "SY01";
```

БД будет создана в каталоге, путь к которому задан в переменной окружения `SY01`, так как `ON` имеет более высокий приоритет, чем `SET DATABASE DIRECTORY`.

В случае некорректного ввода имени переменной окружения в команде `ON`

```
SET DATABASE DIRECTORY "D:\linter\db\TEST_DB\";
CREATE DATABASE "Database Name" ON "D:\TEST_DB1";
```

будет выдана ошибка

```
%GENDB-E-UNDVAR,неопределенная переменная : D:\T
```

```
MAXTAB <количество таблиц>
```

Предполагаемое максимальное количество таблиц в создаваемой БД (по умолчанию 8176). Заданное значение выравнивается с избытком до кратного 1022. Максимально допустимое значение равно 65398.

Примечания

1. Значение данного параметра в существующей БД можно получить, выполнив команду `show $$$SYSRL` утилиты `inl`. Предельный `ROWID` есть значение `MAXTAB`.
2. Максимальное число таблиц в создаваемой БД определяется как (`<заданное пользователем значение MAXTAB>` + 10), округлённое в большую сторону до числа, делящегося без остатка на 1022.
3. После достижения БД указанного значения `MAXTAB` создание новых таблиц прекращается. При необходимости расширения БД можно воспользоваться утилитой `testdb` и увеличить установленные лимиты. Подробное описание и команды

приведены в пункте «Расширение системных таблиц» документа «Тестирование базы данных».

`MAXCOL <количество столбцов>`

Суммарное число всех столбцов во всех таблицах создаваемой БД (по умолчанию 32704). Заданное значение выравнивается с избытком до кратного 1022. Максимально допустимое значение данного параметра равно 65408. После достижения БД этого значения создание новых таблиц прекращается (см. примечание к параметру MAXTAB).

Примечание
Значение данного параметра в существующей БД можно получить, выполнив команду `show $$$ATTRI` утилиты `inl`. Предельный `ROWID` есть значение `MAXCOL`.

`MAXUSR <количество пользователей>`

Максимальное число записей по правам доступа к таблицам (по умолчанию 1022).

Системная таблица `$$$USR` автоматически расширяется по мере необходимости.

`MAXROWSIZE <максимальный размер записи>`

Максимальный размер записи в таблице (значение от 1 до 65535). Реальная длина записи (в байтах) вычисляется путем округления заданного размера до значения, кратного 4096 в большую сторону. Если размер записи получается равным 65536, то он усекается до 65535. Значение по умолчанию 65535.

Если значение <максимального размера записи> больше 65535, то выдается предупреждающее сообщение, а размер записи берется равным 65535.

Примечание
Изменение максимального размера записей в БД после её создания осуществляется с помощью команды `ALTER DATABASE SET RECORD SIZE LIMIT` (см. документ «Справочник по SQL», пункт «Ограничение длины записи»).

`CERTIFIED | ORDINARY`

При указании опции `CERTIFIED` - будет создана сертифицированная БД. Будут активированы следующие проверки: контроль целостности файлов БД, контроль длины пароля пользователя 8 символов, максимальное количество неудачных вводов пароля 4 раза. Опция применяется по умолчанию.

При указании опции `ORDINARY` - БД будет создана без дополнительных проверок.

Примечание
Опции поддерживаются со сборки 6.0.20.1.

В процессе выполнения команды `CREATE` в БД создаются следующие таблицы словаря БД и рабочие файлы:

- `1.01` и `1.11` – таблица таблиц (`$$$SYSRL`);
- `2.01` и `2.11` – таблица атрибутов (`$$$ATTRI`);
- `3.01` и `3.11` – таблица пользователей (`$$$USR`);

- 1.31 – файл бит-векторов (мнемоника SYSWBV);
- 1.41 – рабочий файл (мнемоника SYSWRK);
- 1.51 – файл сортировки (мнемоника SYSSRT);
- 1.61 – специальный файл системного журнала, содержащий информацию о контрольных точках, горячем резервировании и т.д.

Примечание
Если при создании БД размер системного журнала не задается, то по умолчанию создается файл размером 8К.

Параметры запуска ядра созданной БД являются стандартными (принятыми по умолчанию). Для их просмотра/изменения предназначены команды GET, SET данной утилиты.

Процесс создания БД протоколируется на терминале (или в файле, если задан параметр MESSAGE OUTPUT). Если указано NO MESSAGE, протоколирование не выполняется.

В зависимости от планируемого использования функциональных возможностей СУБД ЛИНТЕР должны быть выполнены конфигурационные sql-файлы, которые создают системные таблицы для поддержки этих возможностей. Список файлов SQL и их функциональных возможностей в таблице 1.

Таблица 1. Список файлов SQL и их функциональных возможностей

Имя файла	Обеспечение функциональных возможностей
arepl.sql	Асинхронная репликация (тиражирование) данных
catalog.sql	Поддержка ODBC-драйвера
catalog_oledb.sql	Поддержка OLEDB
cerrors.sql	Создание пустой таблицы кодов завершения ядра СУБД ЛИНТЕР. Загрузка в таблицу собственно кодов завершения и соответствующих им текстов выполняется ядром СУБД ЛИНТЕР из файла errors.lod
charsets.sql	Однобайтовые кодировки
cstables.sql	Словарь кодировок (версии 6.x и выше)
distr.sql	Распределенная обработка данных
enaacc.sql	Доступ работы станциям с БД из дистрибутива и права на тестовые таблицы
events.sql	Управление событиями
geo_cat.sql	Поддержка OpenGIS
inkernel.sql	Для работы системы резервирования данных
mbcps.sql	Многобайтовые кодировки
ora_cat.sql	Поддержка совместимости с СУБД Oracle
pbcat.sql	Поддержка PowerBuilder
search.sql, default.sql	Полнотекстовый поиск
security.sql, extsec.sql	Средства расширенной защиты информации

Имя файла	Обеспечение функциональных возможностей
`systab.sql`	Триггеры, хранимые процедуры и последовательности



Примечание

Если в/на каталоге/устройстве, предназначенном для новой БД, уже содержится некоторая БД, то утилита выдаст запрос-предупреждение о возникновении подобной ситуации. При этом будет выдано имя уже существующей БД и запрос на продолжение операции. В случае отмены операции утилита не будет выполнять процедуру создания БД. В противном случае поверх рабочих файлов старой БД будут переписаны файлы новой. Это разрушит старую БД, т.к. при этой операции не все файлы старой БД будут удалены – все файлы пользовательских таблиц останутся (если они были расположены в данном каталоге), что может привести в дальнейшем к некорректной работе новой БД. Во избежание подобных ситуаций не рекомендуется проводить создание новой БД поверх старой.

Если в команде создания БД указан параметр `NO QUERY`, то запрос на перезапись файлов выдан не будет, что может привести к последствиям, описанным выше.

Чтобы разрешить удаленный доступ клиентских приложений к данной БД, необходимо выполнить команду:

```
grant access on unlisted station to all;
```

(разрешить доступ к этой БД со всех компьютеров)

либо команду на создание станции (разрешить доступ к этой БД только с конкретных компьютеров) (см. документ «Администрирование комплекса средств защиты данных», пункт «Создание/удаление сетевой станции»).

Примеры

1) Создание БД, местоположение которой задает переменная окружения по умолчанию `SY00`.

Шаг 1. Установка средствами ОС переменной окружения `SY00`:

```
SY00=d:\linter\db
```

Шаг 2. Создание БД:

```
gendb>
gendb>create database;
```

2) Создание БД, местоположение которой задается непосредственно в утилите путем указания полного пути к каталогу:

```
gendb>set database dir "d:\linter\db";
gendb>create database;
```

3) Создание нескольких БД, местоположение которых задается непосредственно в утилите путем указания переменной окружения.

Шаг 1. Создать и инициализировать средствами ОС переменные окружения `DB01`, `DB02`:

```
set DB01=d:\linter\db\order
```

```
set DB02=d:\linter\db\sale
```

Шаг 2. Создание БД:

```
gendb>set database path DB01;
gendb>create database "Orders";
gendb>set da pat DB02;
gendb>create database "Sales";
```

4) Создание БД, местоположение которой задается локально переменной окружения.

Шаг 1. Создать и инициализировать средствами ОС переменную окружения TMP0:

```
set tmp0=d:\tempdb
```

Шаг 2. Создание БД:

```
gendb>create database "TestDB" on "TMP0";
```

5) Создание защищенной БД:

```
создать каталог C:\Linter\db\TESTDB
Установить значение переменной SY00: SET SY00=C:\Linter\db\TESTDB
Запустить gendb: gendb
Установить пароль для защищенной БД: gendb> SET PROTECTED DATABASE
 PASSWORD "$des$12345";
Создать БД: gendb> CREATE DATABASE "TESTDB";
Завершить утилиту gendb: gendb> exit
Запустить ядро СУБД: linter64 /pass=$des$12345
```

Конфигурирование БД

Конфигурирование БД (установка параметров запуска ядра СУБД ЛИНТЕР) должно выполняться после создания БД. Если конфигурирование не выполнялось, СУБД будет использовать значения по умолчанию. Конфигурационные параметры (параметры запуска) – значения, хранящиеся в системной БД и извлекаемые (при загрузке) ядром СУБД ЛИНТЕР для настройки своей работы. Эти параметры влияют на общую производительность СУБД.

К ним относятся:

- размеры очередей таблиц, столбцов, каналов, описаний файлов, пользователей;
- число записей и число индексов, просматриваемых системой без прерывания при обработке одного запроса;
- размеры (в том числе максимальные) и количество рабочих файлов и файлов журнала;
- различные режимы обработки изменений, произведенных транзакцией в БД.

Команда «Конфигурирование БД» включает в себя три функции:

1) определение местоположения БД, для которой выполняется конфигурирование и/или просмотр уже установленных параметров конфигурирования;
2) собственно конфигурирование БД;
3) просмотр установленных параметров конфигурирования.

Синтаксис

```
<конфигурирование БД>::=
[<пароль защиты БД>]
[<путь к БД>]
[<установка параметров конфигурирования>]
<просмотр параметров конфигурирования>

<пароль защиты БД>
```

См. пункт Пароль защищенной БД.

```
<путь к БД>
```

См. пункт Путь к БД.

Установка параметров конфигурирования

Синтаксис

```
<установка параметров конфигурирования>::=
SET
[OUTPUT "<имя файла>"]
[DATABASE NAME "<имя БД>"]
[DATABASE DEVICE <переменная БД>]
[SYSWRK DEVICE <переменная рабочего файла>]
[SYSWBV DEVICE <переменная файла бит-векторов>]
[SYSSRT DEVICE <переменная файла сортировки>]
[SYSLOG DEVICE <переменная системного журнала>]
[SYSWRK SIZE <размер рабочего файла>]
[SYSWBV SIZE <размер файла бит-векторов>]
[SYSSRT SIZE <размер файла сортировки>]
[SYSLOG SIZE <размер файла системного журнала>]
[SYSWRK LIMIT <максимальный размер рабочего файла>]
[SYSWBV LIMIT <максимальный размер файла бит-векторов>]
[SYSSRT LIMIT<максимальный размер файла сортировки>]
[SYSSRT COUNT<количество файлов сортировки>]
[SYSLOG COUNT <количество файлов системного журнала>]
[TABLES <длина очереди таблиц>]
[COLUMNS <длина очереди столбцов>]
[FILES <длина очереди файлов>]
[USERS <длина очереди пользователей>]
[CHANNELS <длина очереди каналов>]
[AUTOCONFIG  {ON | OFF}]
[TIMEQUANT {ON|OFF}]
[TQPERIOD <квант обработки запросов>]
[QUANTROW <квант обработки записей>]
[QUANTIND <квант обработки индексов>]
[LOGFILE {ALL | <номер файла>} SIZE <размер>]
[EXTSIZE {AUTO | <размер расширения файлов>}]
```

```
[TRUETYPECOMMIT {ON | OFF}]
[PHRASE INDEX LIMIT<количество фразовых индексов>]
[SQL USERS <количество пользователей SQL>]
[SQL COLUMNS <количество столбцов SQL>]
[SQL PROCEDURES <количество хранимых процедур SQL>]
[SQL CHARSETS <количество кодировок SQL>]
[SQL TABLES <количество таблиц SQL>]
[IN-MEMORY TABLES <количество таблиц «в памяти»>]
[IN-MEMORY COLUMNS <количество столбцов «в памяти»>]
[IN-MEMORY FILES <количество файлов таблиц «в памяти»>]
[WORKAREA COUNT <количество дескрипторов>]
[QUERYCACHE <размер кэша текстов запросов>]
[ANSWERCACHE <размер кэша результатов запросов>]
[CHANMEMLIMIT <максимальный размер памяти каналов>]
[SYSLOG MAX FILES <максимальное количество файлов журнала>]
[SYSLOG RESERVED FILES <количество резервных файлов журнала>]
;
```

Примечание
Если список конфигурационных параметров пуст, команда выдает текущее значение всех параметров на консоль терминала или в заданный файл (см. пункт Просмотр параметров конфигурирования).

`OUTPUT "<имя файла>"`

Определяет текстовый файл, в который будет выводиться информация о параметрах запуска ядра СУБД. Команда `SET` без задания `OUTPUT` выводит информацию на экран терминала.

`DATABASE NAME "<имя БД>"`

Изменяет имя БД.

`SYSWRK DEVICE <переменная рабочего файла>`

Переменная среды окружения, которая специфицирует устройство/каталог рабочего файла `SYSWRK` (файл `1.41`).

`SYSWBV DEVICE <переменная файла бит-векторов>`

Переменная среды окружения, которая специфицирует устройство/каталог рабочего файла `SYSWBV` (файл `1.31`).

`SYSSRT DEVICE <переменная файла сортировки>`

Переменная среды окружения, которая специфицирует устройство/каталог рабочих файлов `SYSSRT` (файл `1.5*`).

`SYSLOG DEVICE <переменная системного журнала>`

Переменная среды окружения, которая специфицирует устройство/каталог файлов системного журнала.

`SYSWRK SIZE <размер рабочего файла>`

Задает размер файла (в страницах). Начальный размер файла по умолчанию 4 страницы.

Примечания

1. Размер файла может быть не больше 2 Гбайт.
2. Если в процессе работы СУБД заданный размер был превышен, то при завершении работы СУБД размер файла будет приведен в соответствие с заданным значением.

`SYSWBV SIZE <размер файла бит-векторов>`

Задает размер файла (в страницах). Начальный размер файла по умолчанию 16 страниц. См. примечания к параметру SYSWRK SIZE.

`SYSSRT SIZE <размер файла сортировки>`

Задает размер файлов сортировки (в страницах). Начальный размер по умолчанию и минимальный размер файла сортировки составляют 2 страницы. См. примечания к параметру SYSWRK SIZE.

`SYSLOG SIZE <размер системного журнала>`

Задает размер файлов системного журнала (в страницах). Максимальное значение 500 000.

`SYSWRK LIMIT <максимальный размер рабочего файла>`

Задает максимальный размер файла `SYSWRK` (файл `1.41`) в страницах. Значение по умолчанию 500 000. Максимальное значение 16777216.

`SYSWBV LIMIT <максимальный размер файла бит-векторов>`

Задает максимальный размер файла `SYSWBV` (файл `1.31`) в страницах. Значение по умолчанию 500 000. Максимальное значение 16777216.

`SYSSRT LIMIT <максимальный размер файла сортировки>`

Задает максимальный размер файлов сортировки (файл `1.5*`) в страницах. Значение по умолчанию 500 000. Максимальное значение 16777216. Расчет необходимого размера файла сортировки по известным параметрам таблицы приведены в «Рекомендации по настройке СУБД ЛИНТЕР», раздел «Расчет размера файла сортировки».

`SYSSRT COUNT <количество файлов сортировки>`

Задает число файлов сортировки.

`SYSLOG COUNT <количество файлов системного журнала>`

Задает число файлов системного журнала.

`TABLES <длина очереди таблиц>`

Задает длину очереди таблиц (по умолчанию 100). Реальная длина очереди таблиц при старте ядра устанавливается на 4 больше заданной. Это обусловлено необходимостью работы с системными таблицами. Минимальное значение 5.

`COLUMNS <длина очереди столбцов>`

Задает длину очереди столбцов (атрибутов) (по умолчанию 500). Реальная длина очереди столбцов при старте ядра устанавливается на 29 больше заданной. Это обусловлено необходимостью работы с атрибутами системных таблиц; не должно быть меньше значения `N=max(Ci)`, где `Ci` – количество столбцов в i-ой таблице БД.

FILES <длина очереди файлов>

Задает длину очереди файлов (по умолчанию 200). Реальная длина очереди файлов при старте ядра устанавливается на 10 больше заданной. Это обусловлено необходимостью работы с системными таблицами. Минимальное значение 5.

USERS <длина очереди пользователей>

Задает длину очереди пользователей (по умолчанию 100). Минимальное значение 1.

CHANNELS <длина очереди каналов>

Задает длину очереди каналов (по умолчанию 1000). Максимальный размер очереди каналов равен 65535.

AUTOCONFIG ON | OFF

Устанавливает режим автоконфигурирования системных очередей и памяти ядра СУБД:

1) в этом режиме ядру СУБД выделяется четверть доступной оперативной памяти компьютера. Каждому процессу сортировки – четверть памяти ядра СУБД. При запуске нескольких ядер СУБД на одном компьютере или при работе с несколькими процессами сортировки может потребоваться отключение данной опции. Значение по умолчанию OFF;
2) память под системные очереди таблиц, файлов, атрибутов и пользователей выделяется автоматически в зависимости от текущих размеров соответствующих системных таблиц;
3) ядро может корректировать размер очереди каналов - например, он не должен быть меньше, чем 10* (число записей в таблице $$$USR), а также может быть увеличен пропорционально размеру пула страниц и обратно пропорционально суммарному размеру системных очередей (но не уменьшен). При этом, в отличие от других настраиваемых автоконфигурацией размеров очередей, размер очереди каналов сохраняется в нулевом кортеже базы (так как при перезапуске базы после падения этот размер должен быть не меньше, чем в упавшем сеансе работы базы, иначе ядро может не стартовать).

Примечания

1. В ОС типа Windows значение по умолчанию OFF, в ОС типа Linux, ЗОСРВ Нейтрино значение по умолчанию ON.
2. Ключи /POOL и /SPOOL имеют больший приоритет.

TIMEQUANT ON | OFF

Разрешает/запрещает использовать при обработке SQL-запросов режим квантования по времени. Значение по умолчанию OFF. Если установлен данный режим, то режим квантования по счетчику (параметры QUANTROW, QUANTIND) отменяется.

TQPERIOD <квант обработки запросов>

Задает величину (в миллисекундах) кванта времени при работе в режиме квантования по времени. Допустимый интервал значений от 1 до 65535. Если установлен режим квантования по времени, а величина кванта времени не задана, используется предыдущее значение кванта времени; если предыдущего кванта времени нет, то по умолчанию квант времени равен 1 миллисекунде.

QUANTROW <квант обработки записей>

Задаёт количество непрерывно обрабатываемых записей (по умолчанию 10).

```
QUANTIND <квант обработки индексов>
```

Задает количество непрерывно обрабатываемых индексных записей (по умолчанию 1).

```
LOGFILE {ALL | <номер файла>} SIZE <размер>
```

Создаёт файл системного журнала заданного порядкового <номера файла> и <размера> в страницах. `ALL` – для случая, когда надо пересоздать все файлы системного журнала.

```
EXTSIZE {AUTO | <размер расширения файлов>}
```

Задает количество страниц для расширения файлов таблиц и рабочих файлов, при превышении установленного начального размера. Допустимый интервал значений от 0 до 8192. Значение по умолчанию `AUTO` (`0`) (расширение выполняется на 1/32 от текущего размера файла с выравниванием до кратного 64 Кбайт, но не больше чем на 32 Мбайт).

Примечание
Опция `AUTO` поддерживается со сборки 6.0.17.92.

Пример

```
SET EXTSIZE AUTO;
SET EXTSIZE 256;
```

```
TRUETYPECOMMIT ON | OFF
```

Задает/отменяет режим немедленного фиксирования изменений, сделанных транзакцией в БД. Значение по умолчанию `ON`.

```
PHRASE INDEX LIMIT <количество фразовых индексов>
```

Задает число одновременно открытых фразовых индексов. Значение по умолчанию 10.

```
SQL USERS <количество пользователей SQL>
```

Задает размер кэша SQL-транслятора при обработке информации о пользователях (владельцах объектов) БД.

```
SQL COLUMNS <количество столбцов SQL>
```

Задает размер кэша SQL-транслятора при обработке информации о столбцах SQL-запроса.

```
SQL PROCEDURES <количество хранимых процедур SQL>
```

Задает размер кэша SQL-транслятора при обработке информации о хранимых процедурах SQL-запроса.

```
SQL CHARSETS <количество кодировок SQL>
```

Задает размер кэша SQL-транслятора при обработке информации о кодовых страницах.

```
SQL TABLES <количество таблиц SQL>
```

Задает размер кэша SQL-транслятора при обработке информации о таблицах SQL-запроса.

```
IN-MEMORY TABLES <количество таблиц «в памяти»>
```

Задает размер системной очереди таблиц «в памяти» ядра СУБД. Значение по умолчанию 0.

```
IN-MEMORY COLUMNS <количество столбцов «в памяти»>
```

Задает размер системной очереди столбцов таблиц «в памяти» ядра СУБД. Значение по умолчанию 0.

IN-MEMORY FILES <количество файлов таблиц «в памяти»>

Задает размер системной очереди файлов, относящихся к таблицам «в памяти» ядра СУБД. Значение по умолчанию 0.

WORKAREA COUNT <количество дескрипторов>

Задает максимальное количество одновременно используемых рабочих областей файла SYSWRK (1.41). Значение по умолчанию 2048. Этот параметр нужно увеличить, если будет получен код завершения 815 (Переполнение конвертера рабочего файла 1.41). Указанное значение округляется до кратного 256.

QUERYCACHE <размер кэша текстов запросов>

Задает количество кэшируемых текстов запросов. Значение в диапазоне от 0 до 65534 включительно. Значение по умолчанию 0, то есть в только что созданной БД кэширование запросов отключено.

Поскольку для хранения кэшируемых запросов используются дескрипторы рабочих областей (задаются командой WORKAREA COUNT, по одному дескриптору на каждый элемент кэша), а количество дескрипторов рабочих областей ограничено (по умолчанию 2048), то при указании количества кэшируемых запросов происходит автоматическое изменение и количества дескрипторов на величину, равную количеству элементов очереди кэшируемых запросов.

ANSWERCACHE <размер кэша результатов запросов>

Задает количество кэшируемых результатов запросов. Значение в диапазоне от 0 до 65534 включительно. Значение по умолчанию 3.

Так как для хранения кэшируемых результатов запросов используются дескрипторы рабочих областей (задаются командой WORKAREA COUNT, по одному дескриптору на каждый элемент кэша), а количество дескрипторов рабочих областей ограничено (по умолчанию 2048), то при указании количества кэшируемых результатов запросов происходит автоматическое изменение и количества дескрипторов на величину, равную количеству элементов очереди кэшируемых результатов запросов.

CHANMEMLIMIT <максимальный размер памяти каналов>

Значение задаётся в байтах, автоматически округляется в большую сторону до ближайшего значения, кратного 4096, и должно находиться в интервале от 65536 (16 страниц по 4 Кбайт) до 16777216 (4096 страниц по 4 Кбайт).

По умолчанию <максимальный размер памяти каналов> равен 1 Мбайт.

SYSLOG MAX FILES <максимальное количество файлов>

Задает максимальное количество файлов системного журнала.

SYSLOG RESERVED FILES <количество резервных файлов>

Задает количество резервных файлов системного журнала.

При запуске ядра СУБД ЛИНТЕР длина очереди файлов увеличивается на 9 + <число файлов сортировки> элементов (все файлы сортировки занимают фиксированные места в очереди).

Примечание
Для получения информации о текущем количестве таблиц в БД необходимо выполнить запрос на подсчет числа записей в системной таблице $$$SYSRL.

Примеры

1) Изменение местоположения рабочего файла БД

   Переменная SY00 используется текущей активной СУБД ЛИНТЕР. Для БД, размещенной на d:\db\bank, задать новое (h:\wrk\bank) местоположение ее рабочего файла.

   - инициализировать средствами ОС переменные окружения:

```
set TMP1= d:\db\bank
set TMP2= h:\wrk\bank
```

   - выполнить команду:

```
gendb>set database dev TMP1 syswrk dev TMP2;
```

2) Изменение параметров защищенной БД

```
Установить значение переменной SY00: SET SY00=C:\Linter\db\TESTDB
Запустить gendb: gendb
Установить пароль для доступа к защищенной БД: SET PROTECTED
 DATABASE PASSWORD "$des$12345";
Изменить требуемый параметр: SET SYSSRT LIMIT 600000;
Проверить установленное значение: GET;
Завершить утилиту: exit
```

Просмотр параметров конфигурирования

Просмотр значений конфигурационных параметров СУБД возможен тремя способами:

1) по команде GET;
2) по команде SET (с пустым списком конфигурационных параметров);
3) с помощью ключа -get (см. пункт Значение параметра конфигурации).

Отличие команд GET и SET – в формате выдаваемой информации.

Назначение

Получение значений параметров запуска ядра СУБД ЛИНТЕР для указанной БД.

Синтаксис

```
<просмотр параметров конфигурирования>::=
[<пароль защиты БД>]
SET [OUTPUT "<имя файла>"] | GET
;
```

Местоположение БД, с которой снимается информация о её параметрах запуска, определяется в соответствии с общим алгоритмом (см. приложение 2).

Нулевые значения параметров кэша транслятора по команде GET:

```
SQL USERS: 0
SQL COLUMNS: 0
```

```
SQL PROCEDURES: 0
SQL CHARSETS: 0
SQL TABLES: 0
```

или по команде SET:

```
Параметры SQL mem 0 col 0 exp 0 lst 0 tab 0.
```

означают, что эти значения явно не заданы при создании БД или не были определены позднее. В этом случае SQL-транслятор использует значение соответствующих параметров ядра СУБД или встроенное значение по умолчанию:

Параметр кэша SQL-транслятора	Используемое значение
SQL USERS	Установленное по команде SET USERS
SQL COLUMNS	Установленное по команде SET COLUMNS
SQL TABLES	Установленное по команде SET COLUMNS
SQL PROCEDURES	20
SQL CHARSETS	10

Значения конфигурационных параметров по команде SET выводятся в следующем формате (пример):

```
                Параметры запуска ядра СУБД
Имя БД 'DEMO DATABASE 'расположение 'SY00' версия 6.0.20
База данных поддерживает сертификационные требования безопасности
Размеры очередей :
1) таблиц          : 100
2) атрибутов       : 500
3) файлов          : 200
4) каналов         : 1000
5) пользователей   : 100
Максимальное количество записей в обработке : 32768
Количество непрерывно обрабатываемых записей : 10
Количество непрерывно обрабатываемых индексных записей : 1
Параметры SQL mem 0 col 0 exp 0 lst 0 tab 0.
Операция "Commit" стандартная.  Разрешен откат очень старых
 транзакций.
Откат очень длинных транзакций запрещен.
Файл SYSWRK расположен <SY00> размером:    9 из 500000 страниц по
 4096 байт
Файл SYSWBV расположен <SY00> размером:    28 из 500000 страниц по
 4096 байт
Файл SYSSRT расположен <SY00> размером:    16 из 500000 страниц по
 4096 байт
Количество страниц для расширения файлов AUTO. Количество
 WORKAREAS 2048
Предельное количество файлов журнала 0
Количество резервных файлов журнала 0
```

```
Журнал Линтер расположен на <SY00>, файлов 5, размер 200 страниц
Журнал Линтер завершен корректно
Кодировка по умолчанию 2
Максимальный размер записи 65535
Лимит одновременно открытых фразовых индексов 10
Ограничения на таблицы в памяти: таблиц = 0, столбцов = 0, файлов
 = 0
Квантование по времени выключено
Размер очереди запросов: 0
Размер очереди результатов запросов: 3
Максимальный размер памяти канала: 262144
```

«Кодировка по умолчанию» устанавливается SQL-запросом `SET DATABASE NAMES <кодировка>` (это кодировка символьных данных в системных таблицах), в частности:

- `0` – кодировка для системных таблиц не задана (по умолчанию используется LATIN1);
- `2` – CP866;
- `3` – KOI-8;
- `4` – CP1251.

(В таком порядке они создаются файлом `dict/charsets.sql`).

Значения конфигурационных параметров по команде `GET` выводятся в следующем формате:

```
<имя параметра>: <значение параметра>
```

`<Имена параметров>` идентичны именам параметров команды `SET`.

Пример вывода по команде `GET`:

```
…
SYSWRK SIZE: 4
SYSWRK LIMIT: 500000
SYSWBV DEVICE: SY00
SYSWBV SIZE: 16
…
```

Примеры

1) Вывод параметров запуска БД в файл.

   Шаг 1. Установить значение переменной `SY00`, задающей местоположение БД:

   например, с помощью операционной системы:

```
set SY00=d:\Linter\db\demo
```

   или с помощью утилиты `gendb`:

```
set database directory "d:\Linter\db\demo"
```

Шаг 2. Выполнить команду gendb:

```
set output "d:\db\config_db.txt";
```

2) Изменение конфигурационного параметра и просмотр его значения.

Переменная SY00 используется текущей активной СУБД ЛИНТЕР. Для БД, размещенной на d:\db\bank, необходимо расширить размер ее очереди таблиц на 30%.

Шаг 1. Инициализировать средствами ОС переменную окружения:

```
set BANK= d:\db\bank
```

Шаг 2. В утилите выполнить следующие команды:

- задать местоположение БД на весь сеанс работы утилиты;
- просмотреть текущие параметры запуска;
- увеличить очередь таблиц;
- просмотреть новые текущие параметры запуска.

```
gendb>set da path "BANK";
gendb>set;
…
Размеры очередей :
1) таблиц          : 100
…
gendb>set  tables 130;
gendb>set;
…
Размеры очередей :
1) таблиц          : 130
…
gendb>exit
```

Справочная информация

Синтаксис команд

Назначение

Вывод на консоль терминала краткой справки о командах утилиты.

Синтаксис

```
<справочная информация>::=HELP
```

Команда HELP выводит общую справку о составе и параметрах команд утилиты. Выводимая на экран информация представляет собой синтаксис обрабатываемых gendb команд:

```
gendb> help
```

```
CREATE DATABASE [<Имя базы>]
[USERNAME <Имя создателя БД>]
[PASSWORD <Пароль создателя БД>]
[MESSAGE OUTPUT <Спецификация файла сообщений>]
[NO MESSAGE]
[NO QUERY]
[ON <Переменная окружения>]
[MAXTAB <Количество таблиц>]
[MAXCOL <Количество столбцов>]
[MAXUSR <Количество пользователей>]
[MAXROWSIZE <Максимальный размер записи>]

%GENDB-I-QUEROW, продолжать ? (YES/NO)[NO]
```

Для продолжения вывода справочной информации командой HELP необходимо ответить YES.

Переменные среды окружения

Назначение

Команда VARIABLE выводит информацию о переменных среды окружения ОС (в том числе и тех, которые могут использоваться утилитой).

Синтаксис

```
<переменные окружения>::=
VARIABLE
[OUTPUT "<имя файла>"]
[<имя переменной> [<имя переменной> ...]];

OUTPUT "<имя файла>"
```

Определяют текстовый файл для вывода информации о переменных среды окружения. По умолчанию информация выводится на экран.

```
<имя переменной>
```

Задает имя переменной среды окружения, справка о которой должна быть получена.

Если не задано ни одно <имя переменной>, по умолчанию выдается значение переменной SY00 (если она определена), в противном случае – путь к каталогу, из которого запущена утилита.

Примеры

1)

```
gendb> variable SY00 DB01;
%GENDB-I-UNDVAR,неопределенная переменная: SY00
%GENDB-I-VARVAL,переменная DB01 указывает на d:\demo
```

2)

```
gendb> variable output "d:\test\var.txt" "Bank";
```

Завершение работы

Назначение

Завершение работы утилиты.

Синтаксис

```
<завершение работы> ::= EXIT
```

Пример

```
gendb>cre database "Test"
_gendb>no message
_gendb>on TMP1
_gendb>maxtab 500
_gendb>;
gendb>exit
```

Ключи

В командной строке обрабатывается только один ключ. Если задано несколько ключей, используется первый из них, остальные игнорируются.

Полная информация о версии СУБД ЛИНТЕР

Синтаксис

```
-version
```

Описание

Ключ предназначен для получения полной информации о версии СУБД ЛИНТЕР, которая может работать с созданной БД.

Пример

```
gendb -version
Linter SQL Bastion v. 6.0.20.5 for LINUX (Socket) 64-bit
Copyright (C) 1990-2025 Relex, Inc. All rights reserved.
```

Краткая информация о версии СУБД ЛИНТЕР

Синтаксис

```
-briefversion
```

Описание

Ключ предназначен для получения краткой информации о версии СУБД ЛИНТЕР, которая может работать с созданной БД.

Пример

```
gendb -briefversion
6.0.17.48
```

Кодовая страница

Синтаксис

```
-ci|-c ["]<имя кодовой страницы>["]
```

Описание

Задает кодовую страницу для интерфейса утилиты. Если ключ не задан, по умолчанию используется язык операционной системы. Если кодовая страница задана неверно или не установлена в ОС, используется англоязычный интерфейс.

Пример

```
gendb cr_db.gdb -ci "cp1251"
```

Значение параметра конфигурации

Синтаксис

```
-get <имя параметра>
```

Описание

Ключ предназначен для получения информации о заданном конфигурационном параметре БД. Имя параметра должно задаваться заглавными буквами. Если имя параметра состоит более, чем из одного слова, то необходимо его заключать в двойные кавычки.

Примечание
Ключ применим только к незащищенной БД

Перед использованием ключа необходимо установить переменную окружения `SY00`, содержащую значение каталога расположения БД, например:

```
set SY00=/usr/linter/db
```

Примечание
В качестве альтернативного способа допускается запуск утилиты `gendb` из подкаталога БД.

Список допустимых имен параметров ключа:

Имя параметра	Имя параметра	Имя параметра
"DATABASE NAME"	"SYSLOG DEVICE"	"TRUETYPECOMMIT"
"DATABASE VERSION"	"SYSLOG COUNT"	"AUTOCONFIG"
"TABLES"	"SYSLOG SIZE"	"SYSWRK DEVICE"

Имя параметра	Имя параметра	Имя параметра
"COLUMNS"	"SYSLOG MAX FILES"	"SYSWRK SIZE"
"FILES"	"SYSLOG RESERVED FILES"	"SYSWRK LIMIT"
"CHANNELS"	"SYSLOG FINISHED"	"SYSWBV DEVICE"
"USERS"	"DEFAULT CHARSET"	"SYSWBV SIZE"
"MAXROWID"	"RECORD SIZE LIMIT"	"SYSWBV LIMIT"
"QUANTROW"	"PHRASE INDEX LIMIT"	"SYSSRT DEVICE"
"QUANTIND"	"IN-MEMORY TABLES"	"SYSSRT SIZE"
"SQL USERS"	"IN-MEMORY COLUMNS"	"SYSSRT LIMIT"
"SQL COLUMNS"	"IN-MEMORY FILES"	"SYSSRT COUNT"
"SQL PROCEDURES"	"PAGE SIZE"	"EXTSIZE"
"SQL CHARSETS"	"TIMEQUANT"	"WORKAREA COUNT"
"SQL TABLES"	"TQPERIOD"	

Сообщения и коды завершения

Информационные сообщения

Информационные сообщения содержат справочную информацию и сведения о текущей выполняемой утилитой работе. Смысл этих сообщений понятен из контекста интерактивного сеанса.

Сообщения об ошибках

В процессе работы с утилитой пользователь может произвести неверные действия. Реакцией утилиты на такие действия служат сообщения об ошибках.

Для анализа ошибок, относящихся к файловым операциям ввода/вывода (конструкция «причина <код ошибки>» в текстах сообщений об ошибках), следует обратиться к эксплуатационной документации ОС.

Ошибочная ситуация может возникнуть и в том случае, если произошёл сбой в работе программы. Такие ситуации также диагностируются и выдаются соответствующие сообщения о возникших ошибках.

Список возможных сообщений об ошибках:

- синтаксическая ошибка;
- ошибка открытия файла <имя файла> на чтение;
- внутренняя ошибка <код ошибки>;
- неопределенная переменная <имя переменной>;
- неверная старшая версия базы данных: <номер версии>;
- неверная младшая версия базы данных: <номер версии>;
- база данных <имя базы> уже существует;
- неверный или поврежденный заголовок базы данных;
- ошибка создания файла <имя файла>, причина <код ошибки>, состояние системы <код системной ошибки>;
- ошибка записи файла <имя файла>, причина <код ошибки>, состояние системы <код системной ошибки>;
- ошибка открытия файла <имя файла> на запись, причина <код ошибки>, состояние системы <код системной ошибки>;
- внутренняя ошибка: недостаточно памяти;
- указан неверный номер журнального файла. Разрешено максимум <количество>;
- указан неверный размер журнального файла. Разрешено максимум <количество>;
- недостаточный размер журнала. Рекомендуется минимум 50 страниц;
- журнал не закончен. При продолжении возможна потеря информации;
- журнал не закончен или неверная версия базы данных. Работа невозможна;
- недопустимый числовой параметр для SET;

- база данных не создана (выдается всегда, когда действия пользователя могут привести к нарушению структуры базы данных).

Коды завершения

Дополнительно к диагностическим сообщениям, выдаваемым на консоль терминала, утилита генерирует программные коды завершения (таблица 2). Проверка программных кодов завершения зависит от операционной системы, в которой запускается утилита, и от способа ее вызова. Так, при вызове утилиты в командных файлах в среде ОС типа Linux, ЗОСРВ Нейтрино код завершения возвращается в переменной `$?`, а в среде ОС Windows – в переменной `errorlevel`.

При вызове утилиты в программных модулях код завершения возвращается в переменных (или структурах данных), определяемых средствами разработки приложений (см. документацию на соответствующие средства разработки).

Таблица 2. Программные коды завершения утилиты gendb

Мнемоническое имя	Числовое значение	Описание
GENDB_RETOK	0	Успешное завершение
GENDB_RETERR	-1	Неизвестная ошибка
GENDB_RET_HEADER_ERR	-2	Неверный или поврежденный заголовок БД
GENDB_RET_VERSION_ERR	-3	В данной версии БД конфигурируемый параметр не поддерживается
GENDB_RET_DB_ERR	-4	Конфигурируемая БД не найдена
GENDB_RET_PARAM_ERR	-5	Неизвестный ключ или параметр команды

Приложение 1

Список ключевых слов и их сокращений

Ключевое слово	Максимально допустимое сокращение
ALL	A
CHANNELS	CHAN
CHARSETS	CHAR
COLUMNS	COL
COUNT	COU
CREATE	CR
DATABASE	DA
DEVICE	DE
DIRECTORY	DI
EXTSIZE	E
FILES	F
GET	G
HELP	H
INDEX	I(i)
LIMIT	LI
LOGFILE	LOG
MAX	MAX
MAXCOL	MAXC
MAXROWSIZE	MAXR
MAXTAB	MAXT
MAXUSR	MAXU
MESSAGE	ME
NAME	NA
NO	NO
OFF	OF
ON	ON
OUTPUT	OU
PASSWORD	PAS
PATH	PAT
PHRASE	PH
PROCEDURES	PR
QUANTIND	QUANTI
QUANTROW	QUANTR
QUERY	QUE
RESERVED	RE
SET	SE
SIZE	SI
SQL	SQ
SYSLOG	SYSL
SYSSRT	SYSS
SYSWBV	SYSW
SYSWRK	SYSWR
TABLES	TA
TRUETYPECOMMIT	TR
USERNAME	USERN
USERS	USERS
VARIABLE	V
WORKAREA	W

Приложение 2
Алгоритм определения местоположения БД

В таблице П2.1 использованы следующие обозначения:

1) столбец `SY00` – переменная окружения по умолчанию определена/не определена в ОС;
2) столбец `set database directory` – перед созданием (конфигурированием) БД в утилите выполнена/не выполнена данная команда (глобальная установка местоположения БД);
3) столбец `set database path` – перед созданием (конфигурированием) БД в утилите выполнена/не выполнена данная команда (глобальная установка местоположения БД);
4) столбец `on (device)` – в соответствующей команде (`CREATE` или `SET`) используется/не используется локальное указание местоположения БД (`create … on`, `set … database device`);
5) знак √ – команда выполнена.

Примечание
Считывание местоположения каталога БД из параметров предыдущего запуска ядра СУБД, хранящихся в реестре ОС, не производится.

Таблица П2.1. Алгоритм определения местоположения создаваемой (конфигурируемой) БД

SY00	set database directory	set database path	on (device)	Местоположение создаваемой (конфигурируемой) БД
–	–	–	–	Каталог выполнения `gendb`
√	–	–	–	Определяется переменной `SY00`
–	√	–	–	Определяется параметром directory
–	–	√	–	Определяется параметром `path`
–	–	–	√	Определяется параметром `ON (DEVICE)`
√	√	–	–	Определяется параметром `directory`
√	–	√	–	Определяется параметром `path`
√	–	–	√	Определяется параметром `ON (DEVICE)`
√	√	–	√	Определяется параметром `ON (DEVICE)`
√	–	√	√	Определяется параметром `ON (DEVICE)`
–	√	–	√	Определяется параметром `ON (DEVICE)`
–	–	√	√	Определяется параметром `ON (DEVICE)`