

**МОБИЛЬНАЯ
РЕЛЯЦИОННАЯ
СУБД**

ЛИНТЕР®

Linter Standard
Linter Bastion
Linter RealTime
Linter Multiversion

**Архивирование и
восстановление базы данных**

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

РЕЛЭКС®

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
Структура документа	2
Принятые обозначения и соглашения	3
Дополнительные документы	4
Условия выполнения утилиты	5
Запуск	5
Команды утилиты	5
Ключи утилиты	7
Ключи управления утилитой	7
Ключи описания объектов	10
Ключи нарастающего архивирования	13
Ключи фразового индекса	13
Ключи поиска устройств при восстановлении БД.....	13
Ключи приоритета.....	14
Ключи просмотра и управления контрольными точками.....	14
Ключи для создания архива на магнитной ленте	15
Ключи для просмотра сохраненных в архиве БД устройств	15
Ключи обработки ошибок.....	15
Выполнение команд утилиты	17
Сохранение базы данных	17
Восстановление базы данных.....	17
Нарастающее архивирование.....	18
Фразовые индексы.....	19
Порядок поиска устройств с архивными файлами.....	20
Создание архива на магнитной ленте.....	21
Проверка архивного файла	21
Пакетный режим работы.....	23
Общие сведения об архивном файле.....	23
Получение справочной информации	23

Оглавление

Автоматизация архивирования базы данных	24
Структура сценария	24
Раздел «Начальные установки»	24
Раздел «Переменные»	26
Раздел «График выполнения»	27
Раздел «Специальные действия»	30
Операторы языка сценария	33
Оператор выражения	33
Комментарий	34
Оператор цикла	34
Оператор условия	34
Оператор запуска утилиты lhb	35
Оператор копирования файла	35
Оператор перемещения файла	36
Оператор переименования файла	36
Оператор удаления файла	36
Оператор создания каталога	37
Оператор синхронного запуска программы	37
Оператор асинхронного запуска программы	37
Оператор окончания программы	37
Оператор прерывания обработки	38
Оператор прерывания обработки	38
Оператор объявления секции обработки ошибок	38
Оператор перехода на обработку ошибки	38
Предопределенные переменные и константы	39
Предопределенные функции	40
Текущее время	40
Номер года	40
Полный номер года	40
Номер месяца	40
Номер дня недели	40
Название дня недели	41
Номер дня месяца	41
Номер текущего часа	41
Номер текущей минуты	41
Информация о файле	42

Выдать сообщение на консоль	42
Поместить сообщение в файл протокола	42
Преобразовать число в строку	42
Ввод строки с консоли	43
Ввод пароля.....	43
Получить значение переменной окружения	43
Пример сценария архивирования	44
Особенности архивирования при параллельной работе с пользователями БД	45
Сообщения утилиты lhb	46
Структура протокола выполнения	46
Протокольные сообщения	47
Оперативное архивирование	50
SQL-запросы оперативного архивирования.....	50
Начать оперативное архивирование	50
Примеры запуска оперативного архивирования	51
Остановить оперативное архивирование	52
Примеры останова оперативного архивирования	53
Механизм асинхронного архивирования	54
Мониторинг процессов асинхронного архивирования.....	55
Коды возврата СУБД ЛИНТЕР при работе резервного сохранения	56
Приложение 1. Коды завершения.....	57
Приложение 2. Примеры применения утилиты.....	59
Указатель ключей	65

Предисловие

Назначение документа

Документ содержит описание утилиты `lhb`, выполняющей резервное сохранение информации из БД ЛИНТЕР в файл архива и, при необходимости, восстановление БД из созданного архива. Эти возможности реализуются на всех программных платформах, на которых функционирует СУБД ЛИНТЕР.

Приводится описание команд, передаваемых через командную строку. В приложениях 1, 2 приведены примеры использования программы резервного сохранения всей базы данных или отдельных ее объектов, примеры восстановления базы данных из файла архива.

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

Для кого предназначен документ

Документ предназначен для системных администраторов и лиц, ответственных за поддержание целостности информации в СУБД ЛИНТЕР.

Утилита может использоваться:

- для полного сохранения базы данных;
- для полного сохранения базы данных с возможностью инкрементного наращивания;
- для последующего полного восстановления базы данных из созданных файлов архива;
- для сохранения отдельных объектов базы данных;
- для восстановления отдельных объектов базы данных;
- утилита поддерживает язык сценариев, т.е. возможно планирование по времени и другим параметрам резервного архивирования базы данных;
- для просмотра и удаления существующих в базе данных контрольных точек;
- для тестирования и листинга файла архива;
- для переноса базы данных между платформами с одинаковой архитектурой.

Необходимые предварительные знания

Для работы с командным интерфейсом необходимо:

- знать основы реляционных баз данных;
- обладать соответствующими правами в СУБД ЛИНТЕР;
- уметь работать в соответствующей операционной системе на уровне простого пользователя.

Структура документа

В разделе «Условия выполнения» показаны условия, необходимые для выполнения утилиты (минимальный состав аппаратных и программных средств, объем свободного места на жестком диске, статус пользователя в системе).

В разделе «Запуск» приведено описание команды запуска утилиты.

В разделе «Команды» приведено краткое описание основных команд, которые обрабатывает утилита lhb.

Раздел «Ключи» включает в себя общее назначение, синтаксис, правила применения и описание уникальных ключей, используемых для передачи параметров утилите в командной строке.

В разделе «Выполнение команд утилиты» приведено назначение, описание применения команд, с помощью которых осуществляются операции по сохранению, архивации данных, тестированию архивных файлов, получению сведений и справочной информации.

Раздел «Автоматизация архивирования базы данных» содержит рекомендации по автоматизации процесса архивации данных, заключающегося в написании сценария выполнения всех операций, которые необходимы при архивировании (восстановлении) конкретной БД.

В раздел «Операторы языка сценария» входит описание, синтаксис и примеры использования операторов специального языка, на котором можно написать командный файл (сценарий) для автоматизированной архивации данных.

В разделе «Предопределенные переменные и константы» приводятся переменные и константы, которые можно использовать в языке, а также их значение.

Раздел «Предопределенные функции» включает в себя описание и синтаксис функций, используемых в языке.

Раздел «Особенности архивирования при параллельной работе с пользователями БД» содержит некоторые важные моменты работы утилиты, которые влияют на степень полноты получаемого архива.

В разделе «Сообщения утилиты lhb» приводится структура диагностического сообщения утилиты, а также возникающие в ходе работы протокольные сообщения, их причины и способы устранения.

Раздел «Оперативное архивирование» описывает возможность полного и инкрементного резервного сохранения базы данных через SQL-запрос самим ядром без участия дополнительных утилит.

Принятые обозначения и соглашения

Обозначение	Пример	Значение
Курсив	<i>Растровым</i> 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
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 -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

Предисловие

Многоточие «...»


Характеристики столбца
MAKE CHAR(20)
MODEL CHAR(20)


Означает, что предшествующая часть может быть повторена любое количество раз

...
SQL>

Многоточие, внутри которого находится запятая «,...»

Указывает на то, 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.

Условия выполнения утилиты

Для работы lhb необходимы следующие условия:

- СУБД ЛИНТЕР должна быть активна (кроме случаев полного восстановления БД);
- в момент запуска lhb СУБД ЛИНТЕР должна иметь в общем случае два свободных канала;
- свободное место на диске для размещения файла(ов) архива, равное примерно половине размера самой СУБД (если предполагается полное сохранение БД с компрессией);
- минимальный объем оперативной памяти – 10 Мбайт;
- пользователь, от чьего имени запускается lhb, должен быть зарегистрирован в БД, к которой осуществляется доступ, и иметь права DBA.

Запуск

Для запуска утилиты необходимо в интерпретаторе команд операционной системы подать командную строку:


```
lhb [<команда> [-<ключ1> [-<ключ2> ...]]]
```

За один сеанс работы утилита обрабатывает только одну команду.


Команды утилиты


Утилита lhb обрабатывает следующие команды:

<u>Команда</u>	<u>Описание</u>
s	Сохранение БД
r	Восстановление БД
l	Справочная информация об архивном файле
t	Тестирование архивного файла
tape	Операции с магнитной лентой
script	Работа по заданному сценарию
set english, set russian	Изменение языка пользовательского интерфейса
sp	Управление контрольными точками
ef<спецификация файла>	Выполнение командного файла


 Команды и ключи (параметры команды, см. ниже) допускается вводить как малыми, так и большими буквами (не допускаются лишь разные регистры букв в одной команде).

 Команды set english, set russian на UNIX-подобных платформах не поддерживаются.

 При вводе значений ключей (имена, пароли, наименования таблиц и т.п.) малые и большие буквы различаются.

 Если задан ключ, не относящийся к команде, то выводится ошибочный ключ, и выполнение утилиты завершается.


Выполнение утилиты

 Если строка местоположения файла (пути к файлу) состоит из 2-х и более слов, то её надо заключать в кавычки.

Ключи утилиты

Ключи управления утилитой

Ключи управления предназначены для обеспечения утилиты `lhb` дополнительной информацией, необходимой при выполнении операций архивирования (восстановления) БД.


Ключ	Описание
<code>-u <имя/пароль></code> <code><имя>/?</code> <code><имя></code>	Задаёт имя и пароль пользователя БД. Если <code><пароль></code> пропущен или задан в виде «?», то он будет запрошен интерактивно. В этом случае вводимое значение пароля на экране будет представлено знаками <code>****</code> . Когда имя и пароль задаются в виде <code><имя/пароль></code> , значение <code><пароля></code> при вводе не скрывается и может быть доступно любому заинтересованному лицу.  Запрос на интерактивный ввод пароля предваряется подсказкой: Введите ваш пароль :
<code>-pf <имя_файла></code>	Задаёт спецификацию (имя и путь) текстового файла с регистрационными данными пользователя. Формат файла: <code>USER=<имя пользователя></code> <code>PASSWORD= <пароль пользователя></code>
<code>-ft <имя_файла></code>	Задаёт спецификацию (имя и путь) текстового файла с описанием сценария архивирования БД (см. стр. 24).
<code>-f <имя_файла></code>	Задаёт спецификацию (имя и путь) архивного файла. Если ключ не задан, по умолчанию архивный файл будет иметь имя <code>db.lhb</code> и размещаться в текущем каталоге.
<code>-fl <имя_файла></code>	Создаёт файл с расширением <code>.log</code> для записи <code>script</code> -операций. Имеет значение только при задании ключа <code>-ft</code> .
<code>-pg</code>	Задаёт постраничный вывод информации на монитор.
<code>-v [<размер></code> <code>[к м]]</code>	Разрешает создание многотомного архива. Параметр <code><размер></code> – целое положительное число, задающее размер архивного тома. Параметр <code>[к м]</code> указывает единицу измерения объёма архивного тома: <code>к</code> – в килобайтах, <code>м</code> – в мегабайтах. Если единица измерения не задана, по умолчанию принимается <code>к</code> . Если <code><размер></code> не задан, то размер архивного тома определяется автоматически. В случае автоматического создания архивных томов файлы архива будут размещаться на магнитном носителе до исчерпания свободного пространства, и только после этого пользователю будет предложено сменить или указать новый магнитный носитель архива. Если ключ не задан, по умолчанию действует режим автоматического создания нового тома.
<code>-c "<комментарий>"</code>	Задаёт комментарий пользователя, сохраняемый в архивном файле.

Ключ	Описание
-g [<code><пароль></code>]	Задаёт пароль для последующего доступа к архиву. Если ключ задан, а <code><пароль></code> не указан, то он будет запрошен интерактивно. Если архив БД создан с паролем, то восстановление БД из этого архива возможно только при предъявлении пароля. Если пароль утерян, то восстановление БД из такого архива невозможно. При вводе <code><пароля></code> регистр клавиатуры не имеет значения.
-p [<code><путь></code>]	Задаёт путь к каталогу, в котором должна быть восстановлена БД. Если <code><путь></code> не указан или ключ не задан, то БД создается в текущем каталоге. Если БД восстанавливается в каталоге, где уже размещена некоторая БД, то при наличии в нём дубликатов восстанавливаемых файлов будет выдаваться запрос на перезапись этих файлов.
-n [<code><имя_сервера></code>]	Задаёт имя удаленного ЛИНТЕР-сервера. Ключ необходим, если выполняется архивирование (восстановление) удаленной БД.
-wait	Ожидать последние изменения в БД (при использовании с командой <code>s</code>) или в архивном файле (при использовании с командой <code>r</code>). Ключ заставляет файл архива находиться в открытом состоянии в ожидании новой архивной информации. Если пользователь нажимает клавиши <code>Ctrl+Break</code> для прекращения работы, то утилита сохраняет (восстанавливает) последний блок файла архива и завершает работу, не дожидаясь поступления изменений от ядра СУБД.

Примеры

1) непрерывное архивирование БД
`lhb s -wait -f arc.lhb ...`
Файл архива будет закрыт только после принудительного прерывания процесса архивации с помощью клавиш `Ctrl+Break`.

2)
`lhb r -wait -f arc.lhb ...`
БД будет восстановлена, но файл архива останется открытым до принудительного прерывания процесса архивации с помощью клавиш `Ctrl+Break`.


 Если одновременно используется один и тот же архивный файл для создания архива в режиме ожидания изменений (в одном каталоге/узле) и восстановления архива (в том же режиме, но в другом каталоге/узле), например, для создания быстро доступной резервной копии БД:

```
lhb s -wait -f arc.lhb ...
lhb r -wait -f arc.lhb ...
```

то процесс, выполняющий восстановление БД, перестанет обрабатывать вновь добавленные в архивный файл записи при достижении конца архивного файла (несмотря на ключ `-wait`).

Это позволяет с помощью клавиш `Ctrl+Break` завершить в подходящий момент процесс восстановления резервной БД и быстро запустить соответствующий ей ЛИНТЕР-сервер.

Ключ	Описание
-movesp	При архивации системного журнала выполнять перенос контрольной точки на текущий файл журнала. Это предотвращает накопление файлов системного журнала при длительной работе утилиты в режиме ожидания изменений. Ключ используется при полном сохранении БД совместно с ключом -wait.
-db	Выполнять синхронизацию данных в архивном файле и в файле БД. Полученный при сохранении блок данных записывается в архивный файл и одновременно разворачивается на диск в виде файлов БД. Ключ используется при полном сохранении БД, фактически эквивалентен совместному использованию режимов сохранения (команда s) и восстановления (команда r) БД.
-dbonly	Выполнять архивирование непосредственно в файлы БД. В процессе архивирования в файл архива записывается только служебная информация, а архивируемые данные сохраняются в БД. Средний объем служебной информации равен примерно 340 байт и может увеличиваться на несколько десятков байт при работе в режиме нарастающего архивирования. Ключ используется при полном сохранении БД.
-wd <fd>	Выполнять периодическую запись символа в заданный номером файловый дескриптор <fd>. В этом режиме при каждой операции обращения к ядру за новой порцией данных производится вывод одного символа. Этот режим может применяться для слежения за работоспособностью lhb (watchdog). В нормальном режиме обращение за новой порцией данных происходит примерно один раз в 30 секунд. Если за 40 - 60 секунд не было выведено ни одного символа, необходимо предпринять действия по рестарту lhb, а возможно, и ядра СУБД ЛИНТЕР после дополнительных проверок. Необходимо также учитывать, что при активном поступлении данных программа слежения должна успевать читать записываемые символы, или переданный lhb файловый дескриптор должен быть переведен в асинхронный режим перед запуском lhb. Недопустимо значение <fd> равное 1 при использовании опций -out, -out+. Ключ используется при полном сохранении БД.
-lr	Выводить сообщения по-русски.
-le	Выводить сообщения по-английски.

 Ключи -lr, -le оказывают действие только на текущий сеанс работы утилиты (в отличие от команд set english и set russian, которые выполняют глобальные установки).

Ключ	Описание
-out <файл>	Задаёт спецификацию файла, в который будут записываться все сообщения утилиты lhb. Если указанный файл уже

<u>Ключ</u>	<u>Описание</u>
	существует, то вся информация из него предварительно удаляется.
-out+ <файл>	Задаёт спецификацию файла, в который будут дописываться все сообщения утилиты lhb. Если указанный файл не существует, то он будет создан.
-notestcrc	Игнорировать BAD CRC.
-version	Предоставить полную информацию о версии утилиты.
briefversion	Предоставить краткую информацию о версии утилиты.

Ключи описания объектов

Ключи описания объектов предназначены для обеспечения утилиты lhb информацией об объектах БД, которые должны быть обработаны при выполнении операций архивирования (восстановления) БД.

При описании ключей этой группы параметр <имя_объекта> определяет маску объекта, к которому относится ключ. Маска может задавать как конкретное имя объекта, (например, таблицу PERSON), так и правило выбора множества объектов аналогично конструкции LIKE в языке SQL (см. документ «СУБД ЛИНТЕР. Описание языка баз данных SQL»). Знак «*» в маске соответствует любому количеству любых символов, знак «?» указывает на один любой символ. Например, «S*» задаёт имена всех объектов, начинающиеся на «S*», «*doc*» - имена всех объектов, у которых в имени в любом месте присутствует набор знаков «*doc*», маска «tab?» определяет имена объектов, имеющих одинаковую длину имени и отличающихся только одним последним знаком – «tab1», «tab2». Допускается задавать несколько масок, разделённых запятой.

Пример

PERSON, AUTO, BANK – конкретные имена таблиц БД, TST* – шаблон имен таблиц, название которых начинается с "TST".


-ot PERSON,AUTO,TST*,BANK


В эту группу входят следующие ключи:

<u>Ключ</u>	<u>Описание</u>
-oall	Сохранение всех объектов БД. Ключ используется для полного логического сохранения БД (т.е. сначала определяются все объекты БД, затем они поочередно архивируются). При полном сохранении без использования этого ключа выполняется физическое (по файлам) сохранение БД. Кроме того, при полном сохранении с ключом -oall возможна потеря данных из архива, полученного при параллельной работе с другими пользователями БД, что исключено при физическом сохранении БД (см. стр. 45).
-ou [<имя_объекта_1> ...]	Задаёт объект БД «Пользователи». Значение <имя_объекта...> должно представлять имя пользователя или маску имен пользователей.
-oe [<имя_объекта_1> ...]	Задаёт объект БД «Хранимые события». Значение

Ключ	Описание
	<имя_объекта...> должно представлять имя хранимого события или маску имен хранимых событий.
-ot [<code><имя_объекта_1> ...</code>]	Задаёт объект БД «Таблицы с данными». Значение <имя_объекта...> должно представлять имя таблицы или маску имен таблиц БД. Существует возможность указывать владельца объекта (например, -ot SYS.AUTO). При использовании этого ключа выполняется полное архивирование таблицы и, соответственно, возможно ее полное восстановление.
-otwd [<code><имя_объекта_1></code>]	Задаёт объект БД «Описание таблицы». Значение <имя_объекта...> должно представлять имя таблицы или маску имен таблиц БД. При использовании этого ключа выполняется архивирование только структуры таблицы.
-oref	Задаёт объект БД «Вторичные ключи» (имеется в виду foreign keys). Ключ используется для сохранения ссылочной целостности БД из архива. Он заставляет утилиту сохранять, помимо собственно данных архивируемой таблицы, и информацию обо всех ссылках этой таблицы на другие таблицы БД. Ключ имеет смысл только при использовании совместно либо с ключом -ot, либо -otwd.
-ov [<code><имя_объекта_1> ...</code>]	Задаёт объект БД «Представления». Значение <имя_объекта...> должно задавать имя представления или маску имен представлений БД.
-os [<code><имя_объекта_1> ...</code>]	Задаёт объект БД «Синонимы». Значение <имя_объекта...> должно представлять имя синонима или маску имен синонимов БД.
-or [<code><имя_объекта_1> ...</code>]	Задаёт объект БД «Роли». Значение <имя_объекта...> должно представлять имя роли или маску имен ролей БД.
-osr [<code><имя_объекта_1> ...</code>]	Задаёт объект БД «Назначения ролей». Значение <имя_объекта...> должно представлять имя пользователя или маску имен пользователей БД, кому назначены роли.
-oa [<code><имя_объекта_1> ...</code>]	Задаёт объект БД «Привилегии». Значение <имя_объекта...> представляет имя таблицы или маску имен таблиц БД, для которых должны быть сохранены (восстановлены) привилегии доступа.
-oap [<code><имя_объекта_1> ...</code>]	Задаёт объект БД «Привилегии на хранимые процедуры». Значение <имя_объекта...> представляет имя хранимой процедуры или маску имен хранимых процедур, для которых должны быть сохранены (восстановлены) привилегии доступа.
-d	Задаёт режим сохранения всех зависимых объектов. Этот ключ заставляет утилиту сохранять (восстанавливать) не только явно заданные объекты,

<u>Ключ</u>	<u>Описание</u>
	но и те, которые связаны с архивируемым объектом (даже если эти связанные объекты не заданы явно как объекты архивирования). Например, если задан этот ключ, то при архивировании некоторой таблицы будут автоматически сохраняться в архиве и все созданные на ее основе представления, все синонимы архивируемой таблицы и т. п. Однако объекты, связанные по правилам ссылочной целостности, автоматически архивироваться не будут. Для этого надо использовать ключ <code>-oref</code> .

 Ключ `-d` совместно с ключом `-oall` игнорируется.

 Данная версия утилиты не определяет объекты, от которых зависит представление (VIEW). В следующих версиях это ограничение будет устранено.

<u>Ключ</u>	<u>Описание</u>
<code>-only <имя_пользователя></code>	Сохранять объекты заданного пользователя.
<code>-otr [<имя_тригг_1> ...]</code>	Задаёт объект БД «Триггеры». Значение <code><имя_тригг...></code> должно представлять имя триггера или маску имен триггеров.
<code>-or [<имя_проц_1> ...]</code>	Задаёт объект БД «Хранимые процедуры». Значение <code><имя_проц...></code> должно представлять имя хранимой процедуры или маску имен хранимых процедур.
<code>-oq [<имя_посл_1> ...]</code>	Задаёт объект БД «Последовательности». Значение <code><имя_посл...></code> должно представлять имя последовательности или маску имен последовательностей.
<code>-on [<имя_сервера_1> ...]</code>	Задаёт объект БД «Сервер репликации». Значение <code><имя_сервер...></code> должно представлять имя сервера репликации или маску имен серверов репликации.
<code>-ob [<имя_правила_1> ...]</code>	Задаёт объект БД «Правила репликации». Значение <code><имя_правила...></code> должно представлять имя правила репликации или маску имен правил репликации.
<code>-takeforeign</code>	Позволяет сохранять чужие объекты (т.е. принадлежащие другим пользователям), к которым разрешен доступ на чтение.
<code>-own</code>	Позволяет восстанавливать как свои объекты (таблицы, представления, процедуры и триггеры), сохраненные другим пользователем. Например, если были предварительно сохранены чужие объекты по ключу <code>-takeforeign</code> , то использование ключа <code>-own</code> позволяет восстанавливать их в БД как свои. Следует с осторожностью пользоваться данным ключом для таких объектов, как представления, процедуры и триггеры, т.к. при восстановлении не производится проверки на существование зависящих от них объектов.

Ключи нарастающего архивирования

Ключи этой группы предназначены для управления нарастающим архивированием БД:

Ключ	Описание
-startinc	Начать режим нарастающего (инкрементного) архивирования.
-inc	Добавить накопленные изменения.
-stopinc	Закончить режим нарастающего (инкрементного) архивирования в указанном архиве.
-vi	Создать новый архивный том.
-fo <спецификация файла>	Указывает местоположение последнего тома инкрементного архива. Новый архивный том будет создаваться в каталоге, указанном в <спецификации файла>. Например, первые тома инкрементного архива были скопированы на отдельный носитель и затем удалены с диска. В этом случае использование данного ключа позволяет продолжить инкрементное архивирование.

Ключи фразового индекса


Ключи этой группы предназначены для сохранения/восстановления фразовых индексов подсистемы полнотекстового поиска СУБД ЛИНТЕР:

Ключ	Описание
-pi	Сохранение в файле архива информации о фразовых индексах для возможности их последующего восстановления. Ключ применим только для полного сохранения БД (без инкрементного).

 Ключ применяется для СУБД ЛИНТЕР версии 5.9 и ниже. Для версии 6.0 и выше он действует по умолчанию.

Ключи поиска устройств при восстановлении БД

Ключи этой группы устанавливают порядок поиска устройства, если при восстановлении БД встречается файл, относящийся к устройству, отличному от заданного в переменной окружения SY00.

 Если устройство задано в переменной SY00, то оно игнорируется, и архив восстанавливается в текущий каталог.

<u>Ключ</u>	<u>Описание</u>
<code>-dp [0 1 2]</code>	Задаёт порядок поиска пути к устройству при восстановлении БД: <ul style="list-style-type: none">• 0 - искать сначала в переменных окружения, затем в списке устройств, сохранённых в файле архива (значение по умолчанию);• 1 – искать сначала в списке устройств, сохранённых в файле архива, затем среди переменных окружения;• 2 - восстанавливать все файлы БД в каталог по умолчанию.

Ключи приоритета

Ключи этой группы предназначены для определения приоритета работы утилиты.

<u>Ключ</u>	<u>Описание</u>
<code>-priority <уровень></code>	Параметр <уровень> может принимать значения: <ul style="list-style-type: none">• HIGH – высокий приоритет (по умолчанию);• NORMAL – средний приоритет;• LOW – низкий приоритет.

Ключи просмотра и управления контрольными точками

При создании инкрементных архивов утилита использует механизм контрольных точек, суть которого в следующем:

- при запуске нарастающего архивирования по команде `startinc` утилита записывает в БД контрольную точку, содержащую необходимую для нарастающего архивирования системную информацию (в частности, дату установки контрольной точки и параметры файлов системного журнала БД, от которых следует производить следующее инкрементное архивирование);
- после установки контрольной точки все файлы системного журнала БД от этой точки навсегда остаются в БД (если же контрольные точки не устанавливаются, то файлы системного журнала, содержащие зафиксированные в БД изменения, удаляются самой СУБД ЛИНТЕР). В связи с этим целесообразно производить удаление уже не используемых старых контрольных точек (например, если начали ещё один инкрементный архив и не хотим продолжать старый).

<u>Ключ</u>	<u>Описание</u>
<code>-list</code>	Показать существующие в БД контрольные точки.
<code>-clear <номер> ALL</code>	Очистить все контрольные точки (параметр ALL) или заданную (<номер>).

 Ключи контрольных точек используются совместно с командой `sr`.

Ключи для создания архива на магнитной ленте

Ключи этой группы предназначены для работы с архивами, хранящимися на магнитных лентах:

Ключ	Описание
<code>-dev <имя_ленты></code>	Монтирование лентопротяжного устройства с системным именем <code><имя_ленты></code> . Допускается одновременное монтирование нескольких устройств.
<code>-list</code>	Показать список файлов на ленте (ключ команды <code>tape</code>).
<code>-crpart <имя_раздела></code>	Создание нового раздела на магнитной ленте (ключ команды <code>tape</code>). Удаляет всю имеющуюся на ней информацию (при переиспользовании ленты) и создает новую метку ленты.

Ключи для просмотра сохраненных в архиве БД устройств

Ключи этой группы предназначены для работы с сохраненными в архиве устройствами.

Ключ	Описание
<code>-ldev</code>	Показать список сохраненных в полном архиве БД устройств (это позволяет заранее определить необходимые переменные окружения для восстановления БД.) Ключ используется в команде <code>L</code> . <code>lhb l -ldev</code>

Ключи обработки ошибок

Ключи этой группы управляют реакцией утилиты `lhb` на ситуации, которые могут возникнуть при ее выполнении. Одновременно можно задавать несколько ключей. Самый высокий приоритет имеет ключ `-qq`, далее `-qs`, затем `-qx`. Если для однотипного события задано несколько ключей, то обработка события выполняется в соответствии со спецификацией высокоприоритетного ключа, остальные ключи в этом случае игнорируются.

 Если ключ задан без параметров, он игнорируется.

Утилита контролирует следующие события:

Событие	Описание
<code>NV</code>	Необходимо создание нового тома.
<code>DF</code>	Создаваемый файл уже существует.

Ключи утилиты

Событие	Описание
CB	Поступил сигнал <Ctrl>+<Break>.
OB	При восстановлении не совпала версия БД.
NB	Слишком новая версия СУБД ЛИНТЕР.
UD	Неизвестное устройство.
ALL	В любом из вышеперечисленных случаев.

Ключ	Описание
-qx [<code><событие_1></code> ...]	Прекратить работу при возникновении указанного события.
-qs [<code><событие_1></code> ...]	Продолжать работу при возникновении указанного события. Нажатие комбинации клавиш Ctrl+Break игнорируется.
-qq [<code><событие_1></code> ...]	Запросить пользователя о дальнейших действиях при возникновении указанного события.

Выполнение команд утилиты

Сохранение базы данных


Под *сохранением БД* понимается выгрузка и сохранение в архивном файле выбранных объектов (возможно, всех, которые есть в БД). Для сохранения БД СУБД ЛИНТЕР должна быть запущена. После сохранения можно восстановить все имеющиеся в ней объекты или любые из них по выбору. Восстановление может быть произведено в уже существующую БД.


Полное сохранение – выгрузка всей БД в архивный файл. Если полное сохранение выполнено без ключа `-oall`, то БД может быть восстановлена только целиком (выделение и восстановление отдельных объектов не допускается). Восстанавливаемая БД будет сформирована заново в указанном каталоге в том виде, в каком была сохранена (запуск ядра СУБД ЛИНТЕР при этом не требуется).

В случае полного сохранения с ключом `-oall` в последующем возможно выборочное восстановление отдельных объектов.

Полное или выборочное сохранение можно выполнять в режиме параллельной работы с другими пользователями БД. Некоторые особенности получаемого в этом случае архивного файла БД описаны на стр. 45.

Сохранение синонимов возможно, если указывается `ИМЯ_ПОЛЬЗОВАТЕЛЯ.ИМЯ_СИНОНИМА`.

 К команде выборочного сохранения относятся ключи: `-u, -f, -ft, -pg, -c, -g, -v, -n, -oall, -ou, -ot, -oref, -ov, -os, -or, -osr, -oa, -otr, -op, -d, -on, -ob, -takeforeign, -only`.

 К команде полного сохранения относятся ключи: `-u, -f, -ft, -pg, -c, -g, -p, -v, -n, -startinc, -inc, -stopinc`.

Примеры

1 Сохранение БД в файле `base.lhb` с комментарием к файлу:

```
lhb s -u SYSTEM/MANAGER -f base.lhb -c "Простой комментарий"
```

2 Сохранение БД в автоматическом режиме по заданному сценарию из файла `time.BSL` с паролем `ABRAKADABRA`:

```
lhb script -u SYSTEM/MANAGER -g ABRAKADABRA -ft time.BSL
```

Восстановление базы данных

Под *восстановлением БД* понимается процесс создания объектов БД, идентичных сохраненным в архивном файле (в случае, если было проведено выборочное сохранение), и создание новой БД, если было проведено полное сохранение. При выборочном восстановлении СУБД ЛИНТЕР должна быть активна. Процесс полного восстановления БД выполняется автономно, и функционирование СУБД ЛИНТЕР не требуется. Выборочное восстановление БД возможно при параллельной работе с другими пользователями БД. Однако в этом случае возможны взаимные помехи в работе, например, при восстановлении данных таблицы, с которой в этот момент работает

Выполнение команд утилиты

пользователь БД (добавляет в нее записи). Особенности восстановления из архива, полученного при параллельной работе с другими пользователями БД, приведены на стр. 45.

Для СУБД версии меньше 5.9, если выполнялось восстановление системных таблиц процедур и триггеров, необходимо перезагрузить ядро. В этом случае выдается соответствующее напоминание. При логическом сохранении этих системных таблиц (\$\$\$PROC, \$\$\$PRCD, \$\$\$TRIG) сохраняется только их структура, а не сами данные. Триггеры и хранимые процедуры, содержащиеся в них, можно сохранить и впоследствии восстановить с помощью ключей `-or` и `-trig`.

Хотя процесс выборочного восстановления возможен при параллельной работе с другими пользователями, во избежание нежелательных последствий рекомендуется осуществлять его только в чрезвычайных ситуациях с соблюдением необходимых мер предосторожности.

К команде восстановления относятся ключи: `-u`, `-f`, `-pg`, `-p`, `-oall`, `-ou`, `-otwd`, `-ot`, `-oref`, `-ov`, `-os`, `-or`, `-osr`, `-oa`, `-d`, `-otr`, `-op`, `-on`, `-ob`, `-own`.

Примеры

1 Полное восстановление БД из архива `base.lhb` в текущем каталоге:

```
lhb r -f base.lhb
```

2 Полное восстановление БД из архива `base.lhb` в каталоге `C:\DB`:

```
lhb r -f base.lhb -p C:\DB
```

Нарастающее архивирование

Под *нарастающим (инкрементным) архивированием БД* понимается специальный режим ведения **полного** архива БД. В отличие от обычного сохранения, которое является разовой работой утилиты архивирования, нарастающее архивирование – синхронный с функционированием БД и продолжающийся во времени процесс, в ходе которого сделанные в БД изменения запоминаются в системном журнале для последующего внесения их в архив. Нарастающее архивирование целесообразно применять в тех случаях, когда регулярное полное сохранение БД является обременительным, а выборочное сохранение затруднено из-за непредсказуемости потока изменений (т.е. трудно или невозможно полностью задать с помощью шаблонов имен перечень изменяемых объектов БД).

Процесс нарастающего архивирования можно разделить на три стадии:

- 1) инициирование нарастающего архивирования – задается ключом `-startinc`. По этому ключу утилита автоматически создает полный архив БД. Этот архив является базовым архивом (контрольной точкой БД, от которой отсчитываются все сделанные в БД изменения). После его создания он получает статус открытого архива, который можно дописывать по мере изменения БД. Открытый архив не должен удаляться или изменяться каким-либо другим способом, кроме как самой утилитой `lhb`. Кроме того, при инициировании нарастающего архивирования устанавливается режим запрета удаления данных из системного журнала БД. Это делается для того, чтобы исключить возможность потери произведенных в БД изменений;
- 2) внесение накопленных изменений в открытый архив задается ключом `-inc`. По этому ключу в открытый архив вносятся все изменения, накопленные в системном журнале (соответственно, в БД) после создания базового архива или предыдущего сеанса обновления открытого архива. По завершении этой операции открытый

архив будет представлять полный и реальный архив БД. После успешного обновления открытого архива системный журнал очищается;

- 3) окончание нарастающего архивирования задается ключом `-stopinc`. По этому ключу никакого сохранения данных не происходит, просто восстанавливается возможность удаления записей системного журнала.

Для системы контрольных точек по команде `stopinc` стирается контрольная точка, соответствующая заданному архивному файлу (необходимо указывать его имя).

При восстановлении БД из нарастающего архива предварительно необходимо внести в открытый архив все накопленные изменения и только потом начать процесс восстановления БД.

Т.к. при нарастающем архивировании устанавливается запрет на удаление из системного журнала, важно правильно определить периодичность обновления открытого архива. При интенсивном изменении БД и редком обновлении открытого архива системный журнал может принять слишком большие размеры (вплоть до исчерпания свободных ресурсов на диске) со всеми вытекающими отсюда последствиями.

К команде сохранения в нарастающем режиме относятся следующие ключи: `-u`, `-f`, `-ft`, `-pg`, `-v`, `-c`, `-g`, `-p`, `-n`, `-startinc`, `-inc`, `-stopinc`.

Примеры

1 Установить режим нарастающего архивирования:

```
lhb s -u SYSTEM/MANAGER -startinc -f base.lhb -c "Нарастающий архив"
```

2 Внести накопленные изменения:

```
lhb s -u SYSTEM/MANAGER -inc -f base.lhb
```

3 Закончить режим нарастающего архивирования:

```
lhb s -u SYSTEM/MANAGER -stopinc -f base.lhb
```

Фразовые индексы

Архивирование информации о фразовых индексах (список таблиц и проиндексированных столбцов) выполняется только при *полном* сохранении БД (без нарастающего архивирования).

Если ключ архивирования фразовых индексов `-ri` задан для любого вида архивирования, кроме полного архивирования БД, он игнорируется.

Восстановление информации о фразовых индексах происходит в конце процесса полного восстановления БД из архива. В результате восстановления информации о фразовых индексах создается файл `phrase.rbd` в том же каталоге, что и файлы БД. Наличие данного файла является сигналом для ядра СУБД на создание фразовых индексов для таблиц и их столбцов, перечисленных в этом файле.

Непосредственное создание фразовых индексов происходит при первом запуске ядра СУБД для работы с восстановленной БД. Во время восстановления фразовых индексов активная работа с ядром невозможна, т.к. механизм квантования еще не запущен (вся работа по восстановлению фразовых индексов производится из функции, отвечающей за инициализацию ядра). О том, что ядро занято построением фразовых индексов, на консоль выдается соответствующее сообщение, содержащее имя таблицы, ее владельца и столбец, для которого создается фразовый индекс.

После успешного создания фразового индекса файл `phrase.rbd` удаляется (при неудачном восстановлении фразовых индексов он сохраняется для возможного последующего применения).

По окончании восстановления фразовых индексов ядро СУБД готово к работе.

Порядок поиска устройств с архивными файлами

При архивировании БД вместе с сохранением собственно данных сохраняется и информация об их местоположении в файловой структуре операционной системы (например, для повышения производительности файлы данных и индексов могут быть размещены на разных внешних устройствах).

В ряде случаев при восстановлении БД из архива может оказаться, что для некоторых файлов требуемых устройств (или каталогов) в файловой структуре операционной системы обнаружить не удалось (например, восстановление БД выполняется на новой аппаратной конфигурации). В такой ситуации утилите необходимы указания о порядке размещения восстанавливаемых файлов. Для этого используется ключ `-dr`.

Поиск устройства для размещения файла восстанавливаемой БД выполняется по следующим правилам:

- 1) если местоположение восстанавливаемого файла соответствует переменной окружения `SY00`, то он размещается соответственно этой переменной окружения;
- 2) если при восстановлении встречается файл, относящийся к устройству, отличному от переменной окружения `SY00`, то поиск пути для определения местоположения файла выполняется следующим образом:
 - если ключ `-dr` не указан или указан без параметра, либо указан с параметром `0`, то производится поиск имени устройства среди переменных окружения;
 - если переменная окружения существует, то считывается значение этой переменной, и на основании его формируется полный путь восстанавливаемого файла. Файл будет восстановлен не в жестко заданном по ключу `-r` пути, а в сформированном;
 - если же переменная окружения, соответствующая имени устройства, не обнаруживается среди переменных окружения, то производится поиск имени устройства в списке устройств из файла архива. Список устройств формируется в самом начале процедуры создания файла архива и содержит информацию об именах устройств и соответствующих им каталогах на дисках;
 - если имя устройства нигде не найдено, то на консоль выдается запрос на размещение восстанавливаемого файла в каталог по умолчанию, т.е. в каталог, из которого была запущена утилита `lhb`, или в каталог, указанный с помощью ключа `-r`;
 - если ключ `-dr` задан с параметром `1`, то порядок поиска устройства будет обратным по сравнению с параметром `0`, т.е. сначала поиск будет осуществляться в списке устройств из файла архива, а затем, если устройство не будет найдено, среди переменных окружения;
 - если ключ `-dr` задан с параметром `2`, то поиск устройств для размещения файлов выполняться не будет, и по умолчанию все файлы восстанавливаемой БД будут

размещаться в каталог по умолчанию. Дополнительные вопросы для подтверждения такого размещения файлов выдаваться не будут.

Создание архива на магнитной ленте

В некоторых случаях (повышенные требования к сохранности и надежности архивных носителей, необходимость ограничения доступа к архивным файлам, технические и экономические ограничения и др.) предпочтительнее создавать и/или хранить архивные файлы БД на магнитных лентах.

Создание архива на магнитной ленте выполняется в следующей последовательности:

- 1) монтирование лентопротяжного устройства. Для этого используется команда `tape` с ключом `-dev`;
- 2) создание раздела на ленте (метки ленты) при первоначальном использовании ленты или при ее переиспользовании. Для этого применяется команда `tape` с ключом `-srpart`;
- 3) создание архивного файла и запись его на ленту. Для этого используется команда сохранения `s` с ключом `-dev`.

Примеры

1 Создание нового архива:

```
lhb tape -dev MT1: -srpart ARCHIVES -c "Для служебного пользования";
```

```
lhb s -u SYSTEM/MANAGER -dev MT1: -f DB_BANK.lhb -c "Банковские расчеты"
```

2 Добавление архива на магнитную ленту:

```
lhb tape -dev MT1:
```

```
lhb s -u SYSTEM/MANAGER -dev MT1: -f DB_STORE.lhb -c "Склад"
```

3 Просмотр содержимого ленты:

```
lhb tape -dev MT1: -list
```

Проверка архивного файла

Под *проверкой (тестированием) архивного файла* понимается контроль целостности и сохранности архива БД. В процессе архивирования утилита, помимо записи сохраняемых объектов в архивный файл, подсчитывает и записывает контрольные суммы этих объектов. При тестировании архива выполняется сверка текущей контрольной суммы каждого заархивированного объекта с эталонной контрольной суммой. Результаты тестирования выдаются в виде протокола, где напротив каждого архивного объекта проставлен результат проверки.



К команде тестирования относятся ключи: `-f`, `-pg`, `-g`.

Пример

Проверка архивного файла `base.lhb` (выборочное архивирование БД).

```
lhb t -f c:\linter\base.lhb
```

Выполнение команд утилиты


Linter On-line BACKUP utility v.4.1.0.893


===Файл c:\linter\base.lhb===


ПРОВЕРКА АРХИВА


ПАРОЛЬ АРХИВА: НЕ ЗАДАН
МЕТОД СЖАТИЯ: STANDARD
ПОЛЬЗОВАТЕЛЬ: SYSTEM
ИМЯ БД: TRIGG
ВЕРСИЯ БД: 6.1
ВЕРСИЯ АРХИВА: 4.1.0
ВРЕМЯ: 10.09.2007 13:45:43 GMT
КОНТРОЛЬНАЯ ТОЧКА:NONE
КОДИРОВКА: WINDOWS
КОММЕНТАРИЙ:

Тип	Имя	Пользователь	Процесс
(-)Table.....	AUTO	SYSTEM	100% OK
(-)Table.....	PERSON	SYSTEM	100% OK
(-)Table.....	\$\$\$COMMENTS	SYSTEM	100% OK
(-)Table.....	ALL_SEQUENCES	SYSTEM	100% OK
(-)Table.....	ALL_PRILTREE	SYSTEM	100% OK
(-)Table.....	M_COMPLECTATION	SYSTEM	100% OK
(-)Table.....	M_CUSTOMERS	SYSTEM	100% OK
(-)Table.....	M_TYPE_DOC	SYSTEM	100% OK
(-)Table.....	M_ED_IZM	SYSTEM	100% OK
(-)Table.....	TYPEINFO	SYSTEM	100% OK
(-)Table.....	PRIV_TYPES	SYSTEM	100% OK
(-)Table.....	\$\$\$PRCD	SYSTEM	100% OK
(-)Table.....	\$\$\$PROC	SYSTEM	100% OK
(-)Table.....	\$\$\$TRIG	SYSTEM	100% OK
(-)Table.....	\$\$\$DEVICE	SYSTEM	100% OK
(-)Table.....	\$\$\$STATION	SYSTEM	100% OK
(-)Table.....	\$\$\$RELATION	SYSTEM	100% OK
(-)Table.....	\$\$\$AUDIT	SYSTEM	100% OK
(-)Table.....	\$\$\$GROUP	SYSTEM	100% OK
(-)Table.....	\$\$\$LEVEL	SYSTEM	100% OK

 Фраза **Метод сжатия** указывает на то, что при создании архива выполняется уплотнение данных.

 Значение, указанное во фразе **Версия архива**, является версией утилиты lhb.

 Столбец **Тип** содержит название объекта. Знак «-» перед названием указывает на то, что объект был сохранен по ключу, знак «+» соответствует зависимому объекту, сохраненному автоматически.

 Столбец **Прогресс** показывает динамику тестирования каждого объекта архива. В процессе тестирования он изменяется от 0 до 100 и информирует пользователя о степени завершенности процесса.

При проверке полного архива БД дополнительно выводятся информационные сообщения, относящиеся к архиву в целом и к контрольным точкам, например:

Тип	Дата	Прогресс
Конец основной секции данных	10.09.2007 13:46:04 GMT	100% OK
Конец добавленной секции	10.09.2007 13:46:04 GMT	100% OK
Конец архива	10.09.2007 13:46:04 GMT	100% OK

Пакетный режим работы

Команда `ef <имя_файла>` позволяет запускать команды утилиты из текстового файла. В файле допустимы переводы строки – они будут восприниматься как пробелы.

Общие сведения об архивном файле

Команда `l` позволяет получить общую информацию об архивном файле.

К команде относятся ключи: `-f`, `-pg`, `-g`.

Пример

Получить общие сведения об архиве.
`lhb l-f c:\linter\base.lhb`

Linter On-line BACKUP utility v.4.1.0.893

===Файл c:\linter\base.lhb===
 СОДЕРЖИМОЕ ФАЙЛА АРХИВА

```

-----
ПАРОЛЬ АРХИВА:  НЕ ЗАДАН
МЕТОД СЖАТИЯ:  STANDARD
ПОЛЬЗОВАТЕЛЬ:  SYSTEM
ИМЯ БД:        DEMO DATABASE
ВЕРСИЯ БД:     6.1
ВЕРСИЯ АРХИВА: 4.1.0
ВРЕМЯ:         10.09.2007 12:33:43 GMT
КОНТРОЛЬНАЯ ТОЧКА:NONE
КОДИРОВКА:     WINDOWS
КОММЕНТАРИЙ:   Дубликат архива БД
    
```

Получение справочной информации

Для получения справочной информации об утилите (список команд и ключей с краткими пояснениями) следует запустить утилиту без командной строки или с ключом `-h`.

Автоматизация архивирования базы данных

Архивирование БД является важной и наиболее часто выполняемой операцией администратора СУБД ЛИНТЕР. Для автоматизации этой работы утилита lhb имеет специальное средство – язык сценария архивирования, на котором можно написать командный файл (сценарий) для выполнения всех операций, необходимых при архивировании (восстановлении) конкретной БД. После того, как сценарий написан, отлажен и передан утилите lhb для исполнения, администратор БД может снять с себя все заботы о сохранности БД.

Сценарий архивирования содержит расписание и порядок выполнения утилитой lhb всего комплекса операций, связанных с ведением архива сложной информационной системы: архивирование БД, создание резервных копий, перенос архива или резервных копий на другие носители данных, удаление устаревших архивов и их копий и др.

При описании языка сценария используется синтаксическая нотация, основанная на формах Бэкуса-Наура (БНФ), со следующими расширениями:

- угловые скобки < > определяют строку, которая именует элемент командной строки (не терминальный символ);
- оператор определения ::= разделяет определяемый элемент (слева от оператора) и собственно его определение (справа от оператора);
- квадратные скобки [] указывают необязательный элемент командной строки или ключа. Часть определения, заключенная в эти скобки, может использоваться, как описано в конструкции, либо пропускаться;
- альтернативный оператор | указывает на то, что синтаксический элемент, следующий за этим символом, является одним из возможных вариантов конструкции;
- многоточие «...» указывает на допустимость повторения синтаксического элемента один или несколько раз внутри синтаксической конструкции;
- ключи командной строки выделены жирным шрифтом;
- пробелы используются для разделения синтаксических элементов.


Структура сценария

Описание сценария может включать следующие разделы:

- 1) начальные установки;
- 2) описание переменных;
- 3) график выполнения;
- 4) специальные действия.

Раздел «Начальные установки»

Раздел «Начальные установки» является необязательным разделом сценария. Он должен включаться в том случае, если в сценарии предполагается использовать данные типа дата/время в формате, отличном от принятого по умолчанию в языке. Установленный разработчиком сценария собственный формат входных данных типа «дата/время» запоминается в предопределенной переменной `TSFORMAT` исходного текста сценария.

 Выходной формат данных типа «дата/время» устанавливается в предопределенной переменной DATEFORMAT.

Описание раздела

SET:

[TSFORMAT = "<входной формат даты/времени>"];

<входной формат даты/времени> ::= <строка формата>

<строка формата> ::=

<элемент формата> [<разделитель><элемент формата> ...]

<разделитель> ::=

/ | - | . | : | , | <символ пробела> | "<символьная константа>"

<элемент формата> ::=

<день месяца>

| <день года>

| <месяц>

| <год>

| <часы>

| <минуты>

<день месяца> ::= DD | dd

<день года> ::= DDD | ddd

<месяц> ::=

MM | mm | MONTH | Month | month | MON | Mon | mon

| MONR

<год> ::= Y [Y [Y [Y]]] | y [y [y [y]]]

<часы> ::= { HH | hh } [{ MID | mid }]

| { HH12 | hh12 } [{ MID | mid }]

| { HH24 | hh24 }

<минуты> ::= MI | mi

Описание шаблонов

Шаблоны DD, dd задают представление дня месяца арабскими цифрами. Диапазон значений этих шаблонов от 1 до 31.

Шаблоны DDD, ddd задают представление дня года арабскими цифрами. Диапазон значений этих шаблонов от 1 до 366.

Шаблоны MM, mm задают представление месяца арабскими цифрами. Диапазон значений этих шаблонов от 1 до 12.

Шаблоны MONTH, Month, month задают полное название месяца.

Шаблоны MON, Mon, mon задают сокращенное (3 символа) название месяца.

Шаблон MONR задает представление месяца римскими цифрами. Допустимые значения этого шаблона: I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII.

Шаблоны Y, y задают представление цифр года арабскими цифрами. Диапазон значений этих шаблонов от 0 до 9.

Шаблоны HH, hh, HH12, hh12 задают представление часа половины суток (до обеда или после обеда) арабскими цифрами. Диапазон значений этих шаблонов от 0 до 11.

Шаблоны HH24, hh24 задают представление часа суток арабскими цифрами. Диапазон значений этих шаблонов от 0 до 23.

Шаблоны MID, mid задают представление символов am, pm.

Шаблоны MI, mi задают представление минут арабскими цифрами. Диапазон значений этих шаблонов от 0 до 59.

По умолчанию используется следующий формат:

```
"DD/MM/YYYY HH24:MI";
```

Примеры форматов и представляемых ими значений

Символьное представление	Формат
16.02.1998:10:35	DD.MM.YYYY:HH:MI
16.02.98:10:35	dd.mm.yy:HH:MI
16-02-98 10:35	DD-mm-YY hh:mi
10 час 30 мин 16/02/1998	HH "час" MI "мин" DD/MM/YYYY
В год 1998 месяца 02 дня 16 10.35	"В год" YYYY "месяца" MM "дня" DD hh.mi
До конца года осталось дней: 264	"До конца года осталось дней: " DDD
10.VII.2001	DD.MONR.YYYY
15/07/1999 07pm:30	DD/MM/YYYY hh12mid:mi
Последняя архивация: март, 23.1998	"Последняя архивация: " MONTH, DD.YYYY

Раздел «Переменные»

Раздел «Переменные» является необязательным разделом сценария. Он должен включаться в том случае, если в исходном тексте сценария предполагается использовать собственные (отличные от предопределенных в языке сценария утилиты) переменные.

Язык сценария допускает два типа переменных - символьные и числовые. Тип переменной явно не объявляется и определяется по типу данных начального значения.

Описание раздела

VARIABLES:

```
[<определение переменной>;
```

```
[<определение переменной>;
```

```
.....
```

```
[<определение переменной>;
```

```
<определение переменной>::=
```

```
  <имя переменной> = <начальное значение>
```

```
<имя переменной>::= <идентификатор>
```

```
<идентификатор>::= алфавитно-цифровая строка длиной не более 18  
знаков.
```

```
<начальное значение> ::=
  "<символьная константа>"
  | <числовая константа>
```

```
<числовая константа> ::= [ + | - ] <целое число>
```



Символьная константа может включать коды \n, \t, \\, \" для представления символа «возврат каретки», «табуляция», непосредственно самого символа \ и символа ".

Примеры

```
FILE_DIR = "c:\linterSQL\db";
```

```
Name_File="";
```

```
Count = 25;
```

Раздел «График выполнения»

Раздел «График выполнения» содержит описание календарных дат и моментов времени, при наступлении которых утилита lhb должна выполнить действия, описанные в сценарии.

Описание раздела

RIGHTS:

```
[<ежечасная операция> [<ежечасная операция> ... ]]
[<ежедневная операция> [<ежедневная операция> ... ]]
[<еженедельная операция> [<еженедельная операция> ... ]]
[<ежемесячная операция> [<ежемесячная операция> ... ]]
[<в заданное время> [<в заданное время> ... ]]
```



Порядок описания операций может быть произвольным.

График выполнения ежечасных операций

Синтаксис

```
<ежечасная операция> ::=
  EVERYHOUR (min=<минуты часа>) <действия>
  <минуты часа> ::= <дата в формате MI>
  <действия> ::= {<блок операторов>}
```


Описание

Указанные <действия> будут выполняться на заданной минуте каждого часа суток. Если к указанному моменту времени утилита выполняет <действия>, заданные каким-либо другим оператором, то выполнение наступивших <действий> будет задержано до полного окончания предыдущих <действий>. В связи с этим при разработке сценария следует так планировать ежечасные <действия>, чтобы утилита успевала выполнять требуемую работу в заданных промежутках времени.


Отрицательное значение аргумента <минуты часа> определяет периодичность повторения заданного действия от начала часа, например,

EVERYHOUR (min=-5) означает, что действие будет выполнено в

..
13:00
13:05
13:10
13:15
..
13:55
14:00
..

 Отрицательное значение аргумента <минуты часа> можно задавать только в виде литерала, а не переменной.

Если значение <минуты часа> не является делителем 60, то промежуток между последним вызовом в предыдущем часе и в начале следующего часа будет меньше этого значения, например, если задано EVERYHOUR (min=-7), то событие вызовется в 13:56, а затем в 14:00.

 **Примечание** (общее для всех операторов данного раздела). При разработке сценария необходимо учитывать следующий важный момент. Если на некоторую календарно-временную дату совпало несколько действий (например, совпали по дате и времени ежедневные и еженедельные операции), то утилита выполнит запланированные действия только одного оператора. Другой при этом будет проигнорирован, причем нельзя определенно сказать, какой именно оператор из нескольких будет выполнен. Поэтому, чтобы избежать таких коллизий, графики выполнения планируемых событий должны отличаться друг от друга хотя бы на 1 минуту.

Пример

```
Everyhour(min=59)
{
  /* Сохранение быстроменяющейся таблицы */
  backup ("s -u SYSTEM/MANAGER -f c:\arc_db -ot Заказы");
exception:
  logprint(ctimestamp()+":\n");
  logprint("\tError=" + TOSTR(CERROR));
  logprint("\tLinError=" + TOSTR(LINERROR));
  logprint("\tSysError=" + TOSTR(SYSEERROR));
  stop;
}
```

График выполнения ежедневных операций

Синтаксис

```
<ежедневная операция> ::=
  EVERYDAY (time=<время суток>) <действия>
  <время суток> ::= '< дата в формате HH24:MI >'
  <действия> ::= {<блок операторов>}
```

Описание

Указанные <действия> будут выполняться каждый день в заданное время суток.

Пример

```
everyday(time='22:00')
```

```
{
/* нарастающее архивирование каждый день */
backup ("s -inc -u SYSTEM/MANAGER -f " +
lastfilename );
}
```

График выполнения еженедельных операций

Синтаксис

<еженедельная операция> ::=
EVERYWEEK (**day**=<день недели>, **time**=<время
суток>) <действия>
<день недели> ::= **Mon** | **Tue** | **Wed** | **Thu** | **Fri** | **Sat** | **Sun**
<время суток> ::= '< дата в формате HH24:MI>'
<действия> ::= {<блок операторов>}



Mon - понедельник, Tue - вторник, Wed - среда, Thu - четверг, Fri - пятница, Sat - суббота,
Sun - воскресенье.

Описание

Указанные <действия> будут выполняться каждую неделю в заданный день недели и время суток.

Пример

```
Everyweek(day=Mon,time='08:00')
{
/* Создание нового архива */
backup ("s -startinc -u SYSTEM/MANAGER -f c:\lhb\"+
tostr(cdate()) + "-" +
tostr(cmonth()) + "-" +
tostr(cyear2()) + "\");
exception:
logprint(ctimestamp()+":\n");
logprint("\tError=" + TOSTR(CERROR));
logprint("\tLinError=" + TOSTR(LINERROR));
logprint("\tSysError=" + TOSTR(SYSERROR));
stop;
}
```

График выполнения ежемесячных операций

Синтаксис

<ежемесячная операция> ::=
EVERYMONTH (**date**=<день месяца>, **time**=<время
суток>) <действия>
<день месяца> ::= <дата в формате DD > | **lastdate**
<время суток> ::= '<дата в формате HH24:MI>'
<действия> ::= {<блок операторов>}
lastdate – последний день текущего месяца

Описание

Указанные <действия> будут выполняться каждый месяц в заданный день месяца и время суток.

Примеры

EVERYMONTH (date=4, time='03:00')

EVERYMONTH (date=lastdate, time='23:59')

Выполнение в заданное время

Синтаксис

<в заданное время> ::=

DIRECT (timestamp=<дата>) <действия>

<дата> – строка даты/времени в соответствии с установленным в разделе SET форматом (или с форматом по умолчанию – в случае отсутствия в сценарии этого раздела).

<действия> ::= {<блок операторов>}

Описание

Указанные <действия> будут выполнены один раз при наступлении заданной даты.

Раздел «Специальные действия»

Раздел «Специальные действия» содержит описание действий, которые утилита lhb должна выполнять в определенные моменты своей работы и/или при возникновении ошибочных ситуаций. Раздел не является обязательным, но если задается, то каждый оператор в нем должен быть представлен в единственном экземпляре.

Описание раздела

SPECIAL:

[<действия в начале работы>]

[<действия в конце работы>]

[<действия при ошибочной ситуации>]

Указанные <действия в начале работы> выполняются программой только в случае успешного запуска, т.е. при условии отсутствия синтаксических ошибок в тексте сценария или в командной строке. Возможны два варианта выполнения <действий в начале работы> в зависимости от используемого сценария:

- 5) сценарий содержит раздел «График выполнения» с указанием циклически осуществляемых операций (ежечасных, ежедневных и т.п.). В этом случае утилита lhb должна функционировать в режиме 24 часа в сутки, 7 дней в неделю, т.е. теоретически «вечно», поэтому <действия в начале работы> (опять-таки теоретически) будут выполнены всего лишь один раз в жизни утилиты. Если это так, то необходимость в программировании и отладке <действий в начале работы> отпадает (один раз эти действия администратор БД может выполнить и вручную). Однако, как показывает практика, реальная жизнь утилиты весьма далека от теоретической модели. Выполнение утилиты может быть прервано по разным причинам: сбой в оборудовании, аварийное завершение утилиты по собственной инициативе из-за неустраняемых ошибок (оператор STOP), прекращение работы по инициативе администратора БД (например, в связи с необходимостью изменения файла сценария) и др. Таким образом, в реальной жизни повторные запуски утилиты lhb являются скорее не исключением, а правилом. В этой ситуации необходимость в разработке <действий в начале работы> становится актуальной, и степень актуальности зависит от частоты перезапуска утилиты lhb.

В <действиях в начале работы> можно выделить следующие обычно задаваемые функции:

- подготовительные операции (например, проверка наличия достаточного дискового пространства для размещения архивного файла, создание необходимых каталогов, возможность монтирования лентопротяжного устройства в случае создания архива на магнитной ленте, инициирование нарастающего архивирования – создание открытого архива и т.п.);
 - проверка результатов предыдущего запуска утилиты lhb и устранение выявленных нарушений в архиве (тестирование, по крайней мере, последней копии архивного файла и повторное создание архива).
- б) сценарий содержит раздел «График выполнения», но в разделе отсутствует описание циклически выполняемых операций (ежечасных, ежедневных и т.п.). В этом случае утилита lhb будет запускаться на выполнение всегда по инициативе администратора БД, и в <действия в начале работы> можно включить операции, которые обычно планируются в циклических операциях, например, удаление архивных файлов с истекшим сроком хранения.

Указанные <действия в конце работы> выполняются утилитой только в случае ее корректного завершения – по команде <CTRL>+<C> или с помощью оператора STOP (инициатива самой утилиты lhb). Если работа утилиты прекращена из-за отказа оборудования или операционной системы, то <действия в конце работы> не выполняются. Это свойство утилиты lhb может быть использовано, в частности, для определения способа завершения предыдущего запуска утилиты. Например, если в <действиях в начале работы> создать метку некоторого файла нулевой длины, а в <действиях в конце работы> удалить этот файл, то присутствие на диске указанного файла при запуске утилиты будет свидетельствовать о ее предыдущем некорректном завершении.

Указанные <действия при ошибочной ситуации> определяют реакцию утилиты в случаях ошибочных ситуаций на глобальном уровне. На данный уровень обработки ошибок утилита выходит, если нет ни одного нижнего уровня, где выявленная ошибка могла быть обработана. Язык сценария предоставляет два способа обработки ошибок: с помощью конструкции iferr, относящейся непосредственно к оператору языка (см. стр. 33) и конструкции exection, объявляющей секцию обработки всех ошибок блока операторов, внутри которой расположена эта секция. В утилите lhb предусмотрена следующая иерархия обработки ошибок:

- 1) если синтаксис оператора предусматривает конструкцию iferr, и эта конструкция задана, то в случае выявления ошибки при выполнении оператора она обрабатывается так, как это предусмотрено в конструкции iferr данного оператора;
- 2) если конструкция iferr не задана или не предусмотрена синтаксисом оператора, ошибка обрабатывается в секции exection текущего блока операторов;
- 3) если текущий блок операторов не содержит секцию обработки ошибок, утилита пытается найти эту секцию на более высоком уровне вложенности блоков операторов. Если на каком-нибудь уровне вложенности секция обработки ошибок будет найдена, то ошибка обрабатывается в соответствии с правилами этой секции;
- 4) если ни одной секции внутри блоков операторов не задано, ошибка будет обработана в глобальной секции (оператор iferr в разделе special);
- 5) если глобальная секция обработки ошибок не определена, то ошибка будет проигнорирована. Последствия распространения этой ошибки на процесс архивирования (восстановления) БД полностью перекладываются на разработчика сценария.

Действия в начале работы

Синтаксис

<действия в начале работы> ::= BEFORE <действия>
<действия> ::= { <блок операторов> }

Описание

Указанные <действия> выполняются в момент запуска утилиты lhb при условии отсутствия синтаксических ошибок в тексте сценария или в командной строке.

Действия в конце работы

Синтаксис

<действия в конце работы> ::= AFTER <действия>
<действия> ::= { <блок операторов> }

Описание

Указанные <действия> выполняются при корректном завершении работы утилиты lhb (в том числе и при завершении по инициативе пользователя по команде <Ctrl>+<C>).

Действия при ошибочной ситуации

Синтаксис

<действия при ошибочной ситуации> ::= IFERR <действия>
<действия> ::= { <блок операторов> }

Описание

Указанные <действия> выполняются в случае выявления ошибки, не обработанной в блоках операторов сценария.

Операторы языка сценария

Оператор выражения

Оператор выражения задает выражение строкового или числового типа. В таблице 1 представлены операции, допустимые для строковых операндов.

Таблица 1. Операции строковых операндов выражения

Операция	Название операции	Тип результата	Примечание
+	Конкатенация	Символьный	
=	Присваивание	Символьный	
==	Сравнение на равенство	Числовой	0 - операнды не равны 1 - операнды равны
!=	Сравнение на неравенство	Числовой	0 - операнды равны 1 - операнды не равны

В таблице 2 указаны допустимые операции для числовых операндов.

Таблица 2. Операции числовых операндов выражения

Операция	Название операции	Примечание
+	Сложение	
-	Вычитание	Бинарная операция
-	Изменение знака	Унарная операция
=	Присваивание	
==	Сравнение на равенство	0 – условие не выполнено, 1 – выполнено
!=	Сравнение на неравенство	0 – условие не выполнено, 1 – выполнено
>	Сравнение на больше	0 – условие не выполнено, 1 – выполнено
<	Сравнение на меньше	0 – условие не выполнено, 1 – выполнено
>=	Сравнение на больше или равно	0 – условие не выполнено, 1 – выполнено
<=	Сравнение на меньше или равно	0 – условие не выполнено, 1 – выполнено
AND	Логическое И	0 – условие не выполнено, 1 – выполнено
OR	Логическое ИЛИ	0 – условие не выполнено, 1 – выполнено
NOT	Логическое отрицание	0 – условие не выполнено, 1 – выполнено

Комментарий

Комментарий может начинаться в любой позиции строки между синтаксическими элементами языка и должен располагаться в одной строке исходного текста сценария. Признаком начала и конца комментария является сочетание символов `/*` и `*/` соответственно.


Оператор цикла

Синтаксис

While (<условие цикла>)

```
{ <блок операторов> }
```

<условие цикла> ::= выражение, результат вычисления которого имеет числовой тип.

 Если <блок операторов> состоит из одного оператора, скобки `{,}` можно не указывать.

Описание

В качестве <условия цикла> может использоваться любое выражение, результатом вычисления которого является значение числового типа.

<Блок операторов> выполняется циклически до тех пор, пока значение <условия цикла> не равно нулю.

Пример

```
while(1)
{
    l=l-1;
    /* нарастающее архивирование */
    backup ("s -inc -u" + myname + "/" + mypass + "
        -f " + "\"" + lastfilename + "\"");
    if(CERROR==0)
        logprint(ctimestamp() + ":" + ">>" +
            lastfilename);
}
```

Оператор условия

Синтаксис

If (<условие выбора>)

```
{ <блок операторов 1> }
```

```
[ else
```

```
{ <блок операторов 2> } ]
```

<условие выбора> ::= выражение, результат вычисления которого имеет числовой тип

Описание

Если значение <условия выбора> не равно нулю, выполняется <блок операторов 1>, в противном случае (если задана конструкция `else`) - <блок операторов 2>.

Примеры

```
if (name == "") /* если пользователь не указал имя и пароль */
{
    MYNAME = GETSTR("Введите имя:");
    MYPASS = GETPASS("Введите пароль:");
}
```

```

    }
  else
  {
    MYNAME = NAME;
    MYPASS = PASSWORD;
  }

```

```

if (not exist (FILES_DIR))
  mkdir (FILES_DIR); /* создать директорий для архива */

```

Оператор запуска утилиты lhb

Синтаксис

```

Backup (<командная строка>);
[iferr{ <обработка ошибки> }
{ <блок операторов 2> } ]
<командная строка> ::= <символьное выражение>

```

Описание

Оператор выполняет запуск утилиты lhb с командами и ключами, заданными в <командной строке>.

Конструкция iferr будет выполнена в случае возникновения ошибки. Если она не задана, то ошибка, выявленная при выполнении оператора, будет обработана в соответствии с иерархией обработки ошибок (см. стр. 32).

Пример

```

backup ("s -startinc -u " + myname + "/" + mypass + "
-f " + "\"" + FILES_DIR + "\" +
tostr(cdate()) + "-" + tostr(cmonth()) + "-" +
tostr(cyear2()) + "\"");

```

Оператор копирования файла

Синтаксис

```

Copy (<спецификация копируемого файла> , <спецификация копии>);
[ iferr{ < блок операторов > }
<спецификация копируемого файла> ::= <символьное выражение>
<спецификация копии> ::= <символьное выражение>

```

Описание

Оператор выполняет копирование заданного файла. <Спецификация копии> задает новое местоположение и имя файла. Имя файла в обеих спецификациях должно быть определено явно, использование знака «*» не допускается. Исходный файл остается без изменений.

Пример

```

copy("25-12-98.lhb","c:\arc_db\25-12-98.lhb");

```

Оператор перемещения файла

Синтаксис

```
Move (<спецификация перемещаемого файла>, <спецификация перемещения>);  
[ iferr { <блок операторов> } ]  
<спецификация перемещаемого файла> ::= <символьное выражение>  
<спецификация перемещения> ::= <символьное выражение>
```

Описание

Оператор выполняет перемещение заданного файла. <Спецификация перемещения> определяет новое местоположение и имя файла. Имя файла в обеих спецификациях должно быть задано явно, использование знака «*» не допускается.

Пример

```
move("25-12-98.lhb", "../old/25-12-98.lhb");
```

Оператор переименования файла

Синтаксис

```
Rename (<спецификация переименоваемого файла>, <спецификация  
перенаименования>);  
[ iferr { <блок операторов> } ]  
<спецификация переименоваемого файла> ::= <символьное выражение>  
<спецификация перенаименования> ::= <символьное выражение>
```

Описание

Оператор выполняет переименование заданного файла. <Спецификация перенаименования> вводит новое имя и тип файла. Местоположение исходного файла не изменяется.

Пример

```
rename("c:\arc_db\25-12-98.lhb", "25-12-98.arc");
```

Оператор удаления файла

Синтаксис

```
Delete (<спецификация файла>);  
[ iferr { <блок операторов> } ]  
<спецификация файла> ::= <символьное выражение>
```

Описание

Оператор выполняет удаление указанного файла. Имя файла в <спецификации файла> должно быть задано явно, использование знака «*» не допускается.

Пример

```
delete("../old/___24-12-98.lhb");
```

Оператор создания каталога

Синтаксис

```
CRDIR (<спецификация каталога>);  
[ iferr { <блок операторов> } ]  
<спецификация каталога> ::= <символьное выражение>
```

Описание

Оператор выполняет создание указанного каталога.

Оператор синхронного запуска программы

Синтаксис

```
Run (<спецификация программы>);  
[ iferr { <блок операторов> } ]  
<спецификация программы> ::= <символьное выражение>
```

Описание

Оператор запускает на выполнение указанную программу, и утилита lhb переходит в ожидание завершения работы этой программы. <Спецификация программы> должна однозначно определять имя программы и ее местонахождение. Если путь к программе не прописан, она будет искаться в текущем каталоге. Результат выполнения программы возвращается в предопределенной переменной CERROR.

Оператор асинхронного запуска программы

Синтаксис

```
Exec (<спецификация программы>);  
[ iferr { <блок операторов> } ]  
<спецификация программы> ::= <символьное выражение>
```

Описание

Оператор запускает на выполнение указанную программу, и утилита lhb продолжает свою работу. <Спецификация программы> должна однозначно определять имя программы и ее местонахождение. Если путь к программе не прописан, она будет искаться в текущем каталоге. Результат выполнения программы не контролируется.

Оператор окончания программы

Синтаксис

```
Stop
```

Описание

Оператор прекращает выполнение утилиты lhb.

Оператор прерывания обработки

Синтаксис

`Break`

Описание

Оператор прекращает работу текущего оператора `if` или выполняет преждевременный выход из цикла `while`.

Оператор прерывания обработки

Синтаксис

`Continue`

Описание

Оператор вызывает выполнение следующей итерации охватывающего цикла (`while`, `if`).

Оператор объявления секции обработки ошибок

Синтаксис

`Exception`

Описание

Оператор объявляет начало секции обработки ошибок текущего уровня вложенности.

Оператор перехода на обработку ошибки

Синтаксис

`Error`

Описание

Оператор выполняет переход в секцию `exception` или в блок операторов `iferr`, соответствующий текущему оператору.

Предопределенные переменные и константы

В языке сценария реализованы следующие предопределенные переменные (т.е. такие переменные, использовать которые можно без их объявления и инициализации):

<u>Имя переменной</u>	<u>Значение переменной</u>
LASTFILENAME	Спецификация пути и имени (без расширения) последнего сохраненного архивного файла.
LASTDATE	Дата последнего сохранения.
CERROR	Код завершения утилиты lhb или программы, выполненной по оператору run. Этот код будет проверяться для вызова iferr.
LINERROR	Код завершения, возвращенный СУБД ЛИНТЕР.
SYSERROR	Код завершения, возвращенный операционной системой.
NAME	Имя пользователя из командной строки.
PASSWORD	Пароль пользователя из командной строки.
FILENAME	Имя файла из командной строки.
DATEFORMAT	Выходной формат данных типа »дата/время«. Элементы формата аналогичны элементам формата в операторе SET: TSFORMAT.

Предопределенные функции

Текущее время

Синтаксис

STIMESTAMP()

Тип возвращаемого значения: символьный.

Описание

Функция возвращает текущую календарную дату в формате предопределенной переменной DATEFORMAT.

Для установки формата, отличного от используемого по умолчанию, необходимо присвоить переменной DATEFORMAT соответствующее значение, например:

```
dateformat="YYMMDD-HH24MI";
```

Номер года

Синтаксис

CYEAR2()

Тип возвращаемого значения: числовой.

Описание

Функция возвращает две последние цифры текущего года.

Полный номер года

Синтаксис

CYEAR4()

Тип возвращаемого значения: числовой.

Описание

Функция возвращает полный номер текущего года.

Номер месяца

Синтаксис

CMONTH ()

Тип возвращаемого значения: числовой.

Описание

Функция возвращает номер текущего месяца (число в диапазоне от 1 до 12).

Номер дня недели

Синтаксис

CDAY()

Тип возвращаемого значения: числовой.

Описание

Функция возвращает номер текущего дня недели (число в диапазоне от 1 до 7, нумерация начинается с понедельника).

Название дня недели

Синтаксис

CWEEKDAY ()

Тип возвращаемого значения: символьный.

Описание

Функция возвращает первые три символа названия текущего дня недели. Названия выдаются в англоязычном виде, например:

Mon, Tue, ...

Номер дня месяца

Синтаксис

CDATE()

Тип возвращаемого значения: числовой.

Описание

Функция возвращает номер текущего дня месяца (число в диапазоне от 1 до 31).

Номер текущего часа

Синтаксис

CHOUR ()

Тип возвращаемого значения: числовой.

Описание

Функция возвращает номер текущего часа (число в диапазоне от 0 до 23).

Номер текущей минуты

Синтаксис

CMIN ()

Тип возвращаемого значения: числовой.

Описание

Функция возвращает номер минуты текущего часа (число в диапазоне от 0 до 59).

Информация о файле

Синтаксис

EXIST(<спецификация файла>)

<спецификация файла> ::= <символьное выражение>

Тип возвращаемого значения: числовой.

Описание

Функция возвращает информацию о существовании указанного файла: 0 - файл не существует, 1 - файл существует.

Выдать сообщение на консоль

Синтаксис

PRINT (<сообщение >)

<сообщение> ::= <символьное выражение>

Тип возвращаемого значения: числовой.

Описание

Функция посылает на консоль (устройство, с которого запущена утилита lhb) текст указанного сообщения. Всегда возвращается значение 1.

Поместить сообщение в файл протокола

Синтаксис

LOGPRINT(<сообщение >)

<сообщение> ::= <символьное выражение>

Тип возвращаемого значения: числовой.

Описание

Функция помещает в файл протокола утилиты указанное сообщение. Если файл протокола не ведется, функция игнорируется. В любом случае возвращается значение 1.

Преобразовать число в строку

Синтаксис

TOSTR(<числовое выражение>)

Тип возвращаемого значения: символьный.

Описание

Функция преобразует числовое выражение в символьное представление.

Ввод строки с консоли

Синтаксис

GETSTR(<символьная переменная>)

Тип возвращаемого значения: символьный.

Описание

Функция осуществляет ввод с консоли символьного текста. Перед вводом данных на консоль выдается подсказка, текст которой содержится в параметре <символьная переменная> вызова функции. Максимальная длина вводимого текста равна 200 символам, признаком конца ввода является переход на новую строку.

Пример

```
GETSTR("\hб>Имя файла архива: ")
```

Ввод пароля

Синтаксис

GETPASS (<символьная переменная>)

Тип возвращаемого значения: символьный.

Описание

Функция осуществляет ввод с консоли символьного текста, представляющего пароль. Вводимые символы визуально отображаются в виде ****. Перед вводом данных на консоль выдается подсказка, текст которой содержится в параметре <символьная переменная> вызова функции. Максимальная длина вводимого текста равна 200 символам, признаком конца ввода является переход на новую строку.

Пример

```
GETPASS("\hб>Пароль ?: ")
```

Получить значение переменной окружения

Синтаксис

GETENV (<имя переменной>)

<имя переменной> ::= <символьное выражение>

Тип возвращаемого значения: символьный.

Описание

Функция возвращает значение заданной переменной окружения, если она существует, и пустую строку – в противном случае.

Пример

```
GETENV ("SY00")
```

Пример сценария архивирования

```
//
// Пример сценария архивации
// -----
Variables:
NUM = 1;
Rights:
Everyday (time = '12:00')
{
  if (CWEEDKDAY() == "Sun") /* Новый архив создавать в
                             воскресенье */
  {
    move (FILENAME + TOSTR(NUM) + ".lhb" , "c:\arc");
    NUM = NUM + 1;
    backup ("s -u " + NAME + "/" + PASSWORD + "
            -f " + FILENAME + TOSTR(NUM) + ".lhb -startinc");
  }
  else { /* В остальные дни ведется нарастающее архивирование* /
    backup ("s -u " + NAME + "/" + PASSWORD + "
            -f " + FILENAME + TOSTR(NUM) + ".lhb -inc");
  } /* end of 'if' */
  Exception: /* Обработка ошибок оператора 'everyday' */
print ("Error=" + TOSTR(CERROR) +
        ",LinError=" + TOSTR(LINERROR) +
        ",SysError=" + TOSTR(SYSERROR));
stop;
}

Special:
before /* just after the start */
{
  print ("Start backup system");
  backup ("s -u " + NAME + "/" + PASSWORD + "
          -f " + FILENAME + TOSTR(NUM) + ".lhb -startinc");
}
after /* after 'stop' or Ctrl-C */
{
  print ("Stop backup system");
  if (ERROR != 0)
    logprint ("Error present:" + TOSTR(ERROR));
}
iferr /* global */
{
  print ("Error=" + TOSTR(CERROR) +
        ",LinError=" + TOSTR(LINERROR) +
        ",SysError=" + TOSTR(SYSERROR));
stop;
}
```

Особенности архивирования при параллельной работе с пользователями БД

Утилита `lhb` позволяет выполнять процесс архивирования (восстановления) данных параллельно с работой пользователей БД. Однако при этом следует учитывать следующие важные моменты, которые влияют на степень полноты получаемого архива и, следовательно, на целесообразность использования этого способа архивирования. В каждом конкретном случае вопрос должен рассматриваться администратором БД индивидуально.

В процессе выборочного архивирования утилита не возвращается к повторному архивированию уже сохраненного и впоследствии измененного до завершения процесса архивирования объекта (т. е. режим «догона» не выполняется). Поэтому, если после архивирования некоторого объекта и до окончания выполнения утилиты этот объект был изменен другими пользователями БД в процессе параллельной работы, новые данные в архив не попадут и, следовательно, при последующем восстановлении будут потеряны.

При полном архивировании БД в параллельном режиме данные будут сохранены полностью.

Если после окончания выборочного или полного архивирования в БД остались некоторые незавершенные транзакции, то при восстановлении из такого архива данные незавершенных транзакций будут потеряны.

Если в процессе выборочного архивирования операция `COMMIT` для некоторой транзакции была выполнена при активной утилите после архивирования объектов этой транзакции и после архивирования системного журнала, то при восстановлении из такого архива изменения, выполненные этой транзакцией, не попадут в файл архива.

Сообщения утилиты lhb

Структура протокола выполнения

Структура диагностического сообщения программы:

```
LINTER HOT BACKUP PROGRAM -- Ver 3.1.0 build 326 -- Relex,Ltd.  
1996-19981
```

<строка протокола выполнения>

...

<строка протокола выполнения>²

lhb return code (<код завершения lhb>) <текст сообщения lhb>³

Linter return code (<код завершения СУБД ЛИНТЕР>) <текст сообщения СУБД
ЛИНТЕР>⁴

System return code (код завершения ОС) <текст сообщения ОС>⁵

Коды завершения и соответствующие тексты сообщений утилиты lhb приведены в приложении 1 данного документа.

Коды завершения и соответствующие тексты сообщений СУБД ЛИНТЕР приведены в документе «СУБД ЛИНТЕР. Справочник кодов завершения».

Коды завершения и соответствующие тексты сообщений операционной системы приведены в эксплуатационной документации на операционную систему. Если операционная система русифицирована, текст сообщения будет на русском языке, в противном случае – на языке операционной системы.

Примеры

```
LINTER HOT BACKUP PROGRAM-Ver 3.1.0 build 326 -Relex,Ltd. 1996-1998  
Ошибка соединения. (1001)
```

lhb return code (30111) - Ошибка соединения с БД
Linter return code (1001)

```
LINTER HOT BACKUP PROGRAM-Ver 3.1.0 build 326 -Relex,Ltd. 1996-1998  
Файл qqq.lhb
```

Не могу открыть файл 'qqq'

lhb return code (30105) - Ошибка открытия файла
System return code (1) - No such file or directory

¹ Строка 1 содержит идентификационные сведения о программе ЛНВ.

² Строки 2 содержат протокольные сообщения программы ЛНВ.

³ Строка 3 содержит код завершения и соответствующий диагностический текст последней операции, которая выполнялась ЛНВ и закончилась неудачно. Описание самой операции находится в последней строке протокола.

⁴ Строка 4 содержит код завершения, сформированный СУБД ЛИНТЕР.

⁵ Строка 5 содержит код завершения, сформированный операционной системой.

Протокольные сообщения

Утилита lhb выдает следующие протокольные сообщения:

1) сообщение **Ошибка соединения**

Причина.

Утилита не смогла установить доступ к БД. Возможны следующие причины:

- БД (локальная или удаленная), которая должна архивироваться (или частично восстанавливаться), не активна;
- было задано неправильное имя удаленного ЛИНТЕР-сервера;
- ошибка сетевого доступа к удаленной БД (например, не загружен один из сетевых драйверов).

Устранение:

- активизировать необходимую БД и повторить выполнение утилиты;
- проверить (уточнить у администратора локальной сети) правильность заданного имени ЛИНТЕР-сервера;
- обратиться к администратору БД или локальной сети для устранения причины отказа сетевого доступа.

2) сообщение **Проверка пароля: строки не идентичны!**

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация закодированного архивного файла, при этом введенный пароль и его подтверждение не совпали.

Устранение:

- указать правильно пароль и его подтверждение (учитывая различие больших и малых букв).

3) сообщение **Не могу открыть файл <имя файла>**

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация несуществующего файла, или пользователь утилиты не обладает необходимыми полномочиями для доступа к архивному файлу.

Устранение:

- задать правильную спецификацию (путь и имя) архивного файла, установить необходимые полномочия пользователю утилиты или изменить атрибуты доступа (в операционной системе) к архивному файлу.

4) сообщение **Неверный архив <имя файла>**

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, который в действительности не является архивом БД.

Устранение:

- указать правильную спецификацию архивного файла.

5) сообщение **Слишком старая версия файла <имя файла>**

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, который создан более ранней версией утилиты lhb.

Устранение:

- для работы с этим архивным файлом использовать соответствующую ему версию утилиты.

6) сообщение Слишком старая версия lhb

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, который создан более поздней версией утилиты lhb.

Устранение:

- для работы с этим архивным файлом использовать соответствующую ему версию утилиты.

7) сообщение неверный пароль

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация закодированного архивного файла, и при запросе пароля был введен неверный пароль.

Устранение:

- попытаться вспомнить (восстановить) пароль. Если восстановить пароль невозможно, архив потерян навсегда.

8) сообщение начать следующий том?

Причина.

Для команды сохранения был задан ключ «Запрашивать разрешение на создание тома» (ключ -qq, значение ключа «Новый том» (NV)).

Устранение:

- ответить ДА, если очередной том многотомного архива нужно создать. При ответе НЕТ процесс архивирования закончится.

9) сообщение недопустимый формат даты <введенная дата>, должен быть 'DD.MM.[YY]YY.NN:MM'

Причина.

В файле сценария строка, задающая дату и время очередного сохранения БД, имеет неправильный формат.

Устранение:

- откорректировать файл сценария сохранения БД.

10) сообщение ошибка сравнения контрольной суммы!

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, целостность которого нарушена.

Устранение:

- ситуация критическая – восстановление из испорченного архива невозможно. Если имеется дубликат архива, использовать его.

11) сообщение файл уже существует. Удалить его?

Причины:

- в команде сохранения БД была задана спецификация существующего архивного файла;
- в команде восстановления БД была задана спецификация существующей БД.

Устранение:

- при ответе да существующий файл будет заменен новым, при ответе НЕТ выполнение утилиты прекратится.

Оперативное архивирование

Полное или нарастающее архивирование БД можно выполнять в оперативном режиме, параллельно с обслуживанием ядром СУБД запросов пользователей. Управление оперативным архивированием осуществляется с помощью соответствующих SQL-запросов. Такое архивирование очень эффективно с точки зрения удаленного администрирования СУБД ЛИНТЕР или при отсутствии возможности (или желания) использовать утилиту lhb. Оперативное восстановление БД ядром СУБД невозможно – для этого необходимо использовать утилиту lhb. Файлы архива, создаваемые утилитой lhb и ядром СУБД при оперативном архивировании, полностью совместимы между собой.

SQL-запросы оперативного архивирования

Начать оперативное архивирование

Функция

Определение оператора запуска процесса архивирования БД в оперативном режиме.

Спецификация

```
<начать архивирование> ::=  
  BACKUP DATABASE [[START | STOP] INCREMENT]  
  [DEVICE <имя устройства>  
  [FILE <спецификация файла> [REWRITE]]  
  [COMMENT <комментарий>]  
  [PASSWORD <пароль>]  
  [VOLUMES <размер тома> [K|M]]  
  [ASYNC];
```

<имя устройства> ::= 4-х символьный литерал

<спецификация файла> ::= символьный литерал

<размер тома> ::= целочисленный литерал

Синтаксические правила

- 1) Опция DATABASE задает полное сохранение БД без возможности нарастающего архивирования;
- 2) <Спецификация файла> задает полную спецификацию (устройство, путь к каталогу и имя) архивного файла. Если тип файла не указан, по умолчанию используется .lhb.

Пример для Unix-платформ:

```
BACKUP DATABASE FILE 'SY00/ARCHIVES/database.lhb';  
BACKUP DATABASE FILE './db.lhb';
```

- 3) Фраза REWRITE разрешает удалять существующий архивный файл и создавать новый с тем же именем;
- 4) Конструкция DATABASE START INCREMENT задает полное сохранение БД с возможностью нарастающего архивирования;
- 5) Конструкция DATABASE INCREMENT – сохранение всех изменений в БД со времени последнего нарастающего сохранения (от последней контрольной точки, созданной, например, с помощью DATABASE START INCREMENT);

- 6) Конструкция `DATABASE STOP INCREMENT` прекращает ведение нарастающего архивирования: последняя контрольная точка удаляется, а архивный файл закрывается. Т. к. по этому SQL-запросу сохранение в архивном файле последней порции накопленных в БД изменений (т. е. начиная с последней контрольной точки) не осуществляется, то перед его выполнением необходимо подать SQL-запрос `BACKUP DATABASE INCREMENT` для исключения потери архивных данных;
- 7) Опция `DEVICE` задает логическое имя устройства, на котором должен создаваться файл архива. Устройство должно быть описано в системной таблице `$$$DEVICE`. Если имя устройства не задано, по умолчанию используется переменная окружения `SY00`. Если `SY00` не определена, архив создается в каталоге запуска ядра СУБД ЛИНТЕР.

Пример
`BACKUP DATABASE DEVICE 'SY02' FILE 'h:\linter\arhiv\sale2804.lhb'`;
- 8) `<Комментарий>` задает текст комментария к архивному файлу;
- 9) `<Пароль>` задает пароль архивного файла;
- 10) Конструкция `VOLUMES <размер тома> [K|M]` разрешает разбивать архивный файл на отдельные тома указанного `<размера тома>`: К – в Кбайтах, М –Мбайтах;
- 11) Опция `ASYNC` задает асинхронное выполнение процесса архивирования.

Общие правила

Если задана опция `ASYNC`, пользователю сразу будет возвращен код завершения, указывающий на результат выполнения SQL-запроса (успешный или неуспешный запуск процесса архивирования). В дальнейшем информацию о текущем состоянии процесса архивирования можно будет получать из системной таблицы `$$$INKERNBACK` (см. раздел «Мониторинг процесса асинхронного архивирования»).

Примеры запуска оперативного архивирования

Пример 1

```
BACKUP DATABASE ASYNC FILE 'DB_Sale.lhb' REWRITE COMMENT 'Оперативный архив от 22.12.2005';
```

Пояснение

Будет запущен процесс оперативного архивирования всей БД в архивный файл «DB_Sale.lhb» (если такой файл уже имеется на диске – он будет предварительно удален без предупреждения), архивный файл будет иметь комментарий «Оперативный архив от 22.12.2005». Процесс будет проходить в асинхронном режиме, т.е. пользователь сразу получит код возврата (0 – если процесс запустился). В дальнейшем процесс архивирования будет протекать независимо от того, работает ли пользователь, запустивший процесс, с БД или нет. Рекомендуется также получить и запомнить идентификатор запущенного процесса для последующего контроля за его состоянием.

Пример 2

```
BACKUP DATABASE FILE 'DB_Sale.lhb' PASSWORD '221205';
```

Оперативное архивирование

Пояснение


Будет запущен процесс синхронного архивирования в архивный файл «DB_Sale.lhb». Если заданный архивный файл уже существует, пользователю будет возвращен соответствующий код завершения, и процесс архивирования завершится; в противном случае процесс архивирования начнется, и пользователь (или клиентское приложение) перейдет в состояние ожидания завершения этого процесса. После окончания процесса архивирования (удачного или неудачного) пользователь получит соответствующий код завершения. Архивный файл будет защищен паролем '221205'.

Пример 3

```
BACKUP DATABASE START INCREMENT FILE 'inc.lhb' VOLUMES 100k;
```

Пояснение

Будет выполнено полное сохранение БД для последующего нарастающего архивирования и в системном журнале установлена контрольная точка, начиная с которой необходимо наращивать инкрементный архив. Файл архива «inc.lhb» будет разделяться на тома размером по 100 Кбайт.

 Если необходимо указать размер в мегабайтах, то следует задавать постфикс «м», например, 1м; если просто в байтах, то постфикс не задается, например, 2000.

Пример 4

```
BACKUP DATABASE INCREMENT FILE 'inc.lhb' VOLUMES 100k;
```

Пояснение

Будет проведена проверка существования в БД контрольной точки нарастающего архивирования, соответствующей файлу архива inc.lhb, с последующим внесением в файл архива накопленных от этой точки изменений. Файлы архива по-прежнему будут разбиваться на тома указанного объема.

Пример 5

```
BACKUP DATABASE STOP INCREMENT FILE 'inc.lhb';
```

Пояснение

Из системного журнала БД будет удалена последняя контрольная точка, соответствующая файлу архива inc.lhb; никакие данные в файл архива не помещаются, процесс архивирования заканчивается с выдачей соответствующего кода завершения.

Остановить оперативное архивирование

Функция

Определение оператора прерывания асинхронного архивирования БД в оперативном режиме.

Спецификация

```
<остановить архивирование>::=  
  BACKUP STOP { ALL | SINCE <дата>}  
  [UNTIL <дата>]  
  | <номер процесса>;
```

<дата> ::= дата-время литерал

<номер процесса> ::= целочисленный литерал

Синтаксические правила

- 1) Опция ALL задает останов всех процессов архивирования (как тех, которые запущены пользователем, подавшим эту команду, так и тех, которые иницированы другими пользователями БД);
- 2) Конструкция SINCE <дата> задает останов процессов архивирования, активизированных после указанной <даты>;
- 3) Конструкция UNTIL <дата> задает останов процессов архивирования, активизированных до указанной <даты>;
- 4) <Номер процесса> – системный идентификатор процесса архивирования. <Номер процесса> возвращается в поле ROWID блока управления запросом CBL при использовании интерфейса нижнего уровня (см. документ «СУБД ЛИНТЕР. Интерфейс нижнего уровня») или в переменной ROWIDPCI_ при использовании встроенного SQL (см. документ «СУБД ЛИНТЕР. Встроенный SQL») в случае успешного запуска процесса архивирования.

Общие правила

- 1) Прервать можно только асинхронное архивирование;
- 2) Прервать процесс архивирования может либо пользователь, который его инициировал, либо пользователь с правами DBA (права DBA позволяют прерывать любые активные процессы архивирования, в том числе и других пользователей).

Примеры останова оперативного архивирования

Пример 1

```
BACKUP STOP 4;
```

Пояснение

Будет произведена попытка останова процесса оперативного архивирования, имеющего идентификатор 4. Сначала ядро СУБД ЛИНТЕР проверит активность указанного процесса (если он уже завершился – будет выдан соответствующий код возврата). Если же процесс оперативного архивирования активен, то выполняются следующие действия:

- закрытие файла архива;
- запись в системную таблицу \$\$\$INKERNBACK информации о причине и стадии (в процентах) завершения процесса;
- останов процесса архивирования.

Пример 2

```
BACKUP STOP ALL;
```

Пояснение

Будет произведена попытка останова всех запущенных в асинхронном режиме процессов оперативного архивирования, в том числе и процессов, запущенных другими пользователями СУБД с правами DBA. В случае успешного выполнения запроса пользователю, подавшему запрос, будет возвращено количество остановленных процессов в поле ROWID блока управления запросом CBL при использовании

интерфейса нижнего уровня (см. документ «СУБД ЛИНТЕР. Интерфейс нижнего уровня») или в переменной ROWIDPCI_ при использовании встроенного SQL (см. документ «СУБД ЛИНТЕР. Встроенный SQL»). Все остановленные процессы завершат свою работу, как описано выше в примере 1 этого раздела.

Пример 3

```
BACKUP STOP ALL SINCE '09.07.2002:12:00' UNTIL '09.07.2002:14:00';
```

Пояснение

Будет произведена попытка останова всех запущенных в указанном диапазоне времени асинхронных процессов оперативного архивирования, в том числе и процессов, запущенных другими пользователями СУБД ЛИНТЕР с правами DBA: будут остановлены процессы, у которых время запуска попадает в интервал между 12:00 и 14:00 девятого июля 2002 года.

Механизм асинхронного архивирования

Асинхронное оперативное архивирование выполняется следующим образом:

- пользователь подает SQL-запрос на запуск оперативного асинхронного архивирования БД и сразу получает от ядра СУБД код завершения об успешном/неуспешном старте процесса;
- запуск последующих процессов асинхронного архивирования можно выполнять по одному и тому же соединению с БД, не дожидаясь завершения предыдущих процессов;
- в случае успешного старта процесса асинхронного архивирования СУБД сама выделяет ему отдельный канал для работы с БД;
- в любом случае (как в случае успешного окончания процесса архивирования, так и неуспешного) этот канал будет корректно закрыт (освобожден) ядром СУБД;
- взаимодействие с выделенным процессу архивирования каналом выполняет ядро СУБД при обработке SQL-запросов BACKUP STOP;

Пользователь БД имеет возможность в любой момент времени получить информацию о ходе выполнения асинхронного архивирования путем выборки нужных данных из системной таблицы \$\$\$INKERNBACK; например, запросом `SELECT * FROM $$$INKERNBACK`. При запуске процесса архивирования в эту таблицу добавляется строка, соответствующая данному процессу, которая в течение архивирования будет периодически модифицироваться для отражения текущего статуса процесса.

- в поле `Status` будет помещено значение в процентах (от 0 до 100) о процессе архивации.

Структура системной таблицы \$\$\$INKERNBACK:

```
CREATE TABLE $$$INKERNBACK (  
  BackId      INTEGER,  
  UserId      INTEGER,  
  ChannelId   INTEGER,  
  LinterRetCode INTEGER,  
  Status      INTEGER,  
  StartTime   DATE,  
  EndTime     DATE);
```

Описание столбцов таблицы:

Столбец	Описание
BackId	Идентификатор процесса архивирования.
UserId	Идентификатор пользователя, подавшего SQL-запрос на оперативное архивирование.
ChannelId	Идентификатор канала, открытого для процесса архивирования (канал, который выделяет само ядро СУБД для порожденного процесса).
LinterRetCode	Код завершения процесса архивирования (см. таблицу).
Status	Число в процентах, отражающее степень завершения процесса архивирования.
StartTime	Дата-время начала процесса архивирования.
EndTime	Дата-время окончания процесса архивирования.

Если эта таблица существует, то добавляется соответствующая строка, содержащая необходимые данные. Наличие таблицы проверяется на этапе начального запуска `in-kernel backup` в асинхронном режиме. В случае ее отсутствия будет выведена соответствующая ошибка, и процесс не запустится. Если таблица существует, то в нее добавляется строка, содержащая `backId`, `userId`, `channelId`, `LinterRetCode` равный нулю, т.к. пока он не известен, `StartTime` и `EndTime`, где `StartTime` будет равно `EndTime` и соответствовать дате-времени старта процесса. По окончании процесса архивации значения `LinterRetCode`, `Status` и `EndTime`, возможно, изменятся в соответствии с действительным кодом завершения процесса, состоянием в процентах и реальным окончанием времени завершения процесса.

Все состояния завершения не асинхронных процессов в эту таблицу попадать не будут, т.к. пользователь получает необходимые данные по окончании процесса архивации.

Мониторинг процессов асинхронного архивирования

Пользователь может в любой момент получить информацию о состоянии процесса оперативного архивирования в статусе резервного сохранения путем SQL-запросов:

```
1 SELECT * FROM $$$INKERBACK;
```

(всех процессов резервного сохранения, запущенных когда-либо)

```
2 SELECT * FROM $$$INKERBACK WHERE backId = number;
```

(конкретного процесса, где `number` – идентификатор процесса, возвращенный после успешного старта оно).

Возвращенным значением станут данные из этой таблицы, где наиболее интересным будет значение поля `Status`. Для него возможны следующие варианты:

- от 0 до 100 – статус, в процентах, работы процесса резервного сохранения;
- 100 – завершение процесса (если значение поля `LinterRetCode` равно 0, то успешное);
- -1 – в случае некорректной или неудавшейся попытки получить статус работающего или завершившегося процесса.

Коды возврата СУБД ЛИНТЕР при работе резервного сохранения

<u>Код</u>	<u>Описание</u>
0	
6931	Таблица \$\$\$INKERNBACK отсутствует в базе данных
6932	Нарушение прав доступа к архивному файлу
6933	Не могу найти файл архива
6934	Превышен предел количества одновременно открытых файлов
6935	Файл архива уже существует
6936	Невозможно открыть файл архива
6937	Невозможно удалить файл архива
6938	Невозможно произвести запись в файл архива
6939	Асинхронный процесс архивации остановлен пользователем
6940	Запрошен останов конкретного процесса - не могу найти указанный процесс
6941	У пользователя нет привилегий DBA
6942	Ошибка чтения из таблицы \$\$\$INKERNBACK
6943	Асинхронный процесс архивации уже остановлен
6944	Невозможно прочесть файл архива
6945	Некорректные инкрементные данные в файле архива

Возможны также другие коды возврата. Нулевой код возврата свидетельствует об успешном завершении.

Приложение 1

Коды завершения утилиты lhb

<u>Код</u>	<u>Причина</u>
30100	Не найдена метка начала ленты
30101	ошибка чтения с магнитной ленты
30102	Ошибка открытия магнитной ленты
30103	ошибка закрытия магнитной ленты
30104	Ошибка записи на магнитную ленту
30105	ошибка открытия файла
30106	Выполнение программы завершено по команде <CTRL>+<C>. Возвращается в операторе After раздела Special
30107	ошибка контрольной суммы
30108	Ошибка чтения файла
30109	ошибка записи в файл
30110	Ошибка удаления файла
30111	ошибка соединения с БД
30112	Ошибка начала сохранения
30113	Невозможно закончить нарастающее архивирование
30114	Не найдена информация о нарастающем архивировании
30115	ошибка записи в нарастающий архив
30116	Ошибка запрещения переиспользования системного журнала
30117	ошибка разрешения переиспользования системного журнала
30118	Ошибка сохранения информации о пользователях БД
30119	ошибка сохранения представления
30120	Ошибка сохранения синонима
30121	ошибка сохранения роли
30122	Ошибка сохранения назначения роли
30123	ошибка сохранения прав доступа
30124	Ошибка сохранения вторичных ключей
30125	ошибка восстановления прав доступа
30126	ошибка восстановления пользователей БД
30127	ошибка восстановления роли
30128	ошибка восстановления назначения роли
30129	ошибка восстановления синонима
30130	ошибка восстановления вторичных ключей
30131	ошибка восстановления представления
30132	ошибка восстановления таблицы
30133	ошибка поиска lhb-раздела на магнитной ленте
30134	файл уже существует на магнитной ленте
30135	ошибка чтения данных с магнитной ленты
30136	Не найдена метка конца магнитной ленты
30137	Слишком старая версия СУБД ЛИНТЕР
30138	Слишком новая версия СУБД ЛИНТЕР
30139	Невозможно переименовать файл
30140	Невозможно создать каталог

<u>Код</u>	<u>Причина</u>
30141	Ошибка запуска программы
30142	Неизвестный метод сжатия блока данных
30143	Версия СУБД ЛИНТЕР не совпадает с версией архива
30144	Неправильно закодированный блок данных
30145	Запорчен файл архива
30146	Недостаточно памяти
30147	Нет данных на магнитной ленте
30148	Ошибка блокирования магнитной ленты
30149	Ошибка разблокирования магнитной ленты
30150	Невозможно найти файл нарастающего архива
30151	Не найдена метка конца магнитной ленты
30152	Файл нарастающего архива не закончен
30153	Ошибка поиска на магнитной ленте
30154	Ошибка установки размера блока на магнитной ленте
30155	Ошибка записи маркера файла
30156	Ошибка поиска маркера файла
30157	Программа не выполняет заданную операцию
30158	Неправильные данные MSBF
30159	Невозможно сохранить представление
30160	Ошибка чтения из устройства ввода
30161	Файл не закончен - корректное восстановление невозможно
30162	Ошибка линтера
30163	Нет свободного места на диске
30164	Не могу сохранить триггер
30165	Не могу сохранить процедуру
30166	Невозможно восстановить триггер
30167	Невозможно восстановить процедуру
30168	Выполнение утилиты завершено пользователем
30169	Ошибка сохранения информации о сервере
30170	Ошибка сохранения правила репликации
30171	Ошибка восстановления информации о сервере
30172	Ошибка восстановления правила репликации
30173	Не могу сохранить событие
30174	Не могу восстановить событие
30175	Ошибка сохранения последовательности
30176	Ошибка восстановления последовательности
30177	Неверные параметры командной строки
30178	Ошибка синтаксиса в скрипте
30179	Ошибка при создании контрольной точки
30180	Ошибка сохранения фразовых индексов
30181	Ошибка сохранения устройств
30182	Контрольная точка не существует в БД
30183	Неверный пароль архива
30184	Ошибка при загрузке сообщений

Приложение 2

Примеры применения утилиты lhb

Создание инкрементного архива и его поддержка

Необходимо нам это для того, чтобы в течение определенного времени иметь наиболее полный резервный архив базы данных, но место на жестком диске ограничено, т.е. мы не можем держать сразу несколько копий полностью сохраненной базы данных.

Пример

Производим запуск инкрементного сохранения:

```
lhb s -u SYSTEM/MANAGER -f db.lhb -startinc
```

В файле db.lhb будет находиться полностью сохраненная база данных, и в самой базе будет установлена контрольная точка, указывающая, с какого места необходимо продолжать инкрементное наращивание. С этого момента все вновь создаваемые файлы журнала также будут оставаться в базе данных (поэтому, если база очень быстро разрастается в объеме, то лучше все-таки воспользоваться периодическим полным сохранением базы).

Далее, по прошествии определенного времени, нам необходимо дописать к файлу архива те изменения, которые произошли в базе со времени последнего инкрементного архивирования:

```
lhb s -u SYSTEM/MANAGER -f db.lhb -inc
```

Файл следует указывать именно тот, для которого мы начинали инкрементный архив. Утилита проверит соответствие контрольной точки, записанной в файл, той, которая находится в базе, и в случае соответствия допишет в конец файла архива те файлы журнала, которые были добавлены за время от последнего сохранения. Контрольная точка в базе будет обновлена для того, чтобы освободить уже ненужные файлы журнала (они впоследствии в ходе работы ядра будут удалены), и следующее инкрементное архивирование будет уже производиться от момента нашего последнего сохранения.

Операцию можно повторять необходимое число раз.

Впоследствии при возникновении каких-либо неприятностей можно будет восстановить данные из этого архива. При старте на восстановленной БД ядро произведет запись всех изменений из журнала в БД, и она будет готова к работе. Если резервное сохранение производилось достаточно регулярно, она будет наиболее близка к изначальной.

По прошествии определенного (возможно длительного) времени необходимо будет начать новый инкрементный архив базы данных. Ведь если база довольно сильно разрослась в объеме за это время, то лучше будет произвести полное или начать новое инкрементное резервное сохранение, т.к. последующее восстановление можно осуществить более быстро, нежели произвести восстановление первоначальной базы и всех накопленных файлов журнала, и предоставить ядру СУБД ЛИНТЕР переносить изменения из файлов журнала в файлы самой БД. При определенных условиях это может потребовать значительного времени.

Исходя из всего вышесказанного, прежде чем начать новое инкрементное сохранение, лучше закрыть старое.

Пример

Если все еще существует файл db.lhb (в который мы накапливали инкрементные изменения), то подаем команду:

```
lhb s -u SYSTEM/MANAGER -f db.lhb -stopinc
```

После чего контрольная точка, соответствовавшая файлу архива, удалится из базы данных. Никакие новые данные при этом в файл архива сохранены не будут. Дальнейшее инкрементное наращивание этого файла станет невозможным.

Другой вариант, как удалить более не используемую контрольную точку (необходимость может возникнуть в случае, если, например, файл архива был поврежден):

```
lhb cp -u SYSTEM/MANAGER -list
```

Результатом будет примерно следующее:

```
Linter On-line BACKUP utility v.4.1.0.870  
for RDBMS Linter SQL v. 6.9.0 for Unix  
Copyright (C) 1995-2002 Relex, Inc. All rights reserved.  
CP(00): Fri Jul 19 10:18:41.04 2002 BACKUP (LongLived) File: 0  
CP(01): Fri Jul 19 10:20:05.19 2002 BACKUP (LongLived) File: 0
```

Т.е. в базе данных установлены две контрольные точки, каждая из которых была создана в результате старта инкрементного архивирования. Допустим, что первая из них нами уже не будет использоваться для инкрементного архивирования, и поэтому вполне логично ее удалить:

```
lhb cp -u SYSTEM/MANAGER -clear 0
```

В результате первая контрольная точка удалится, а файлы журнала освободятся, и если они нигде больше не нужны, будут удалены.

Создание полного архива базы данных

Преимущества использования метода полного архивирования данных:

- наиболее быстрое последующее восстановление базы данных из файла архива;
- сохранение информации о фразовых индексах поддерживается только для полного сохранения;
- не используются контрольные точки, и, как следствие, не остаются более не используемые файлы журнала.

Недостатки:

- относительно длительный процесс архивации;
- т.к. база данных может достигать значительного размера, то хранение нескольких файлов архива может потребовать немало места;
- по вышеуказанным причинам относительно затруднительно вести частое резервное сохранение базы данных.

Поэтому метод, применяемый при резервном архивировании, должен выбираться системным администратором в каждом случае индивидуально. Наиболее оптимальный – комбинирующий методы полного резервного сохранения и инкрементного.

Пример

Необходимо произвести полное сохранение базы данных в файл full.lhb, который в свою очередь следует разбить на тома по 650Мб (например, для хранения на CD). Хотелось бы, чтобы утилита не выдавала запрос на создание нового тома и чтобы сохранилась информация об имеющихся фразовых индексах. Подаем следующую команду:

```
lhb s -u SYSTEM/MANAGER -f full.lhb -v 650M -qc NV -pi
```

Другой вариант запуска: имеется текстовый файл run.txt, в котором написана эта строка:

```
s -u SYSTEM/MANAGER -f full.lhb -v 650M -qc NV -pi
```

Тогда подаем команду читать параметры из внешнего файла:

```
lhb ef run.txt
```

Результат будет аналогичен предыдущему.

Восстановление из полного архива базы данных

При восстановлении не имеет значения, какое было сохранение базы данных – полное или инкрементное.

Ядро СУБД ЛИНТЕР может быть не запущено.

Перед восстановлением желательно создать/очистить каталог, в который будет производиться восстановление базы (настоятельно рекомендуется, чтобы восстанавливаемый каталог не был каталогом существующей базы данных). Также желательно убедиться в наличии свободного места на диске, исходя из того, что его может потребоваться в два-три раза больше, чем занимает сам файл архива, из-за используемой в нем компрессии данных.

Пример

Подаем команду на восстановление (предположим, что файл архива был разбит на тома, и мы не хотим подтверждать открытие каждого нового тома) в каталог NEW_DB:

```
lhb r -u SYSTEM/MANAGER -f full.lhb -qc NV -p NEW_DB
```

В результате все файлы восстановленной базы данных будут помещены в каталог NEW_DB. Можно производить запуск ядра СУБД ЛИНТЕР на восстановленной БД.

Если восстановленная база содержала информацию о фразовых индексах, то при старте ядро проанализирует эту информацию и создаст заново необходимые фразовые индексы (что может потребовать определенного времени). Затем база будет готова к работе.

Сохранение и восстановление отдельных объектов БД из архива БД

Сохранение отдельных объектов отличается от полного сохранения структурой создаваемого файла архива. При сохранении отдельных объектов файлы журнала не сохраняются.

Для восстановления объектов необходимо работающее ядро СУБД ЛИНТЕР на уже существующей базе данных. Также желательно, чтобы восстанавливаемые объекты отсутствовали в БД.

При сохранении объектов сохраняется их структура и для таблиц, также сохраняются данные (если не указан ключ `-otwd` при сохранении таблиц).

Случаи необходимости сохранения и восстановления отдельных объектов БД:

- какой-либо из объектов БД очень часто меняется по сравнению с другими (например, таблица), и поэтому ее периодическое сохранение более выгодно, чем создание полного архива базы данных или инкрементного;
- для переноса в другую БД некоторых объектов (например, таблиц и их триггеров, пользователей и ролей);
- при сильной нехватке свободного места на диске.

Пример

Допустим, необходимо сохранить таблицу `CUSTOMER` со всеми ее зависимыми объектами и внешними ссылками. Файл архива `customer.lhb` зашифруем по паролю `password` и снабдим комментарием, что это таблица `CUSTOMER`.

```
lhb s -u SYSTEM/MANAGER -f customer.lhb -g password -c "CUSTOMER table" -ot CUSTOMER -oref -d
```

Для последующего восстановления данной таблицы в БД вводим команду:

```
lhb r -u SYSTEM/MANAGER -f customer.lhb
```

Утилита предложит ввести пароль и попытается восстановить таблицу в базу данных. Если таблица с таким именем уже существует у данного пользователя, то будет выдано соответствующее предупреждение.

Другой возможный вариант: у пользователя `BOLT` есть таблица `CLIENTS`, и он желает передать ее со всем содержимым пользователю `SYSTEM`. Для этого пользователь `BOLT` сохраняет данную таблицу в файл:

```
lhb s -u BOLT/ITALIC -f clients.lhb -ot CLIENTS,
```

и передает созданный файл архива пользователю `SYSTEM`, а тот уже восстанавливает эту таблицу как свою по команде:

```
lhb s -u SYSTEM/MANAGER -f clients.lhb -ot CLIENTS -own
```

Следует всегда в подобных случаях стараться переносить объекты, от которых эта таблица зависит.

Пример использования языка сценариев для проведения регулярного инкрементного архивирования данных

Преимущества от использования языка сценариев для архивирования данных: можно гибко настроить график сохранения по конкретным временным условиям в определенные файлы архива.

Пример

Пример листинга файла сценариев, который позволяет сохранять базу данных при запуске и затем каждый день в 02:00; при этом предыдущие файлы переименовываются соответственно в `arc1.lhb...arc4.lhb`. Свежий файл имеет имя `db.lhb`. Если при запуске `lhb` задать ключ `-fl FILE.LOG`, то история сохранения будет накапливаться в файле `FILE.LOG`. Содержимое файла `script.bsl`:

```
/* ----- */
```

```

Variables:
  USERNAME   ="SYSTEM"; /* user name */
  USERPASSWORD ="MANAGER"; /* user password */
  ARCDEVICE  = "."; /* for new files */
  ARCFNAME   = ""; /* new name for old file */
  CHKSUF     = ".lhb"; /* suffix for checkpoint file */
  NUMFILE    = 1;
/* ----- */
Rights:
Everyday (time = '02:00')
{
  NUMFILE = 1;
  while (NUMFILE < 5)
  {
    if (exist (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
              ".lhb"))
    {
      if (NUMFILE == 1)
        delete (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
              ".lhb");
      Else
        rename (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
              ".lhb",
              ARCDEVICE+"arc" + TOSTR(NUMFILE-1) +
              ".lhb");
    } /* if */
    NUMFILE = NUMFILE + 1;
  } /* while */
  rename (ARCDEVICE+"db.lhb", ARCDEVICE+"arc" +
        TOSTR(NUMFILE-1) + ".lhb");
  backup ("s -u "+USERNAME+"/"+USERPASSWORD+"
        -f "+ARCDEVICE+"db.lhb"+" -qc DF");
  logprint (CTIMESTAMP() + "---File" + "db" + CHKSUF +
        "created.\n");
Exception: /* for everyday */
print ("Error=" + TOSTR(CERROR) +
      ",LinError=" + TOSTR(LINERROR) +
      ",SysError=" + TOSTR(SYSERROR));
logprint (CTIMESTAMP() + "---Error=" +
      TOSTR(CERROR) +
      ",LinError=" + TOSTR(LINERROR) +
      ",SysError=" + TOSTR(SYSERROR));
stop;
} /* Everyday */
/* ----- */
Special:
before /* just after the start */
{
  NUMFILE = 1;
  while (NUMFILE < 5)
  {
    if (exist (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
              ".lhb" ))
    {
      if (NUMFILE == 1)
        delete (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
              ".lhb" );
      Else
        rename (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
              ".lhb",
              ARCDEVICE+"arc" + TOSTR(NUMFILE-1) + ".lhb" );
    } /* if */
    NUMFILE = NUMFILE + 1;
  }
}

```

```
    } /* while */
    rename (ARCDEVICE+"db.lhb", ARCDEVICE+"arc" +
        TOSTR(NUMFILE-1) + ".lhb");
    backup ("s -u "+USERNAME+"/"+USERPASSWORD+
        -f "+ARCDEVICE+"db.lhb"+" -qc DF");
    logprint (CTIMESTAMP() + "---File" + "db"+ CHKSUF +
        "created.\n");
}
after /* after stop or Ctrl-C */
{
    print ("---Stop backup system");
    if (CERROR != 0)
        logprint (CTIMESTAMP() + "---Error present:" +
            TOSTR(CERROR));
    logprint (CTIMESTAMP() + "---Stop backup system\n");
}
iferr /* global */
{
    print ("Error=" + TOSTR(CERROR) +
        ",LinError=" + TOSTR(LINERROR) +
        ",SysError=" + TOSTR(SYSERROR));
    logprint (CTIMESTAMP() + "---Error=" +
        TOSTR(CERROR) +
        ",LinError=" + TOSTR(LINERROR) +
        ",LinError=" + TOSTR(LINERROR));
    stop;
}
/* ----- */
```

Для запуска данного файла сценария вводим команду:

```
lhb script -ft script.bsl -fl FILE.LOG
```

Утилита выполнит полное сохранение БД и затем будет ожидать 2:00, после чего повторит очередное полное сохранение в новый файл архива.

Указатель ключей

-briefversion, 11
-c, 7
-clear, 15
-crpart, 16
-d, 12
-db, 10
-dbonly, 10
-dev, 16
-dp, 15
-f, 7
-fl, 7
-ft, 7
-g, 8
-h, 24
-inc, 14
-ldev, 16
-le, 10
-list, 15, 16
-lr, 10
-movecp, 9
-n, 8
-notesterc, 10
-oa, 12
-oall, 11
-oap, 12
-ob, 13
-oe, 11
-on, 13
-only, 13
-op, 13
-oq, 13
-or, 12
-oref, 12
-os, 12
-osr, 12
-ot, 11
-otr, 13
-otwd, 12
-ou, 11
-out, 10
-out+, 10
-ov, 12
-own, 13
-p, 8
-pf, 7
-pg, 7
-pi, 14
-priority, 15
-qc, 17
-qq, 17
-qx, 17
-startinc, 14
-stopinc, 14
-takeforeign, 13
-u, 7
-v, 7
-version, 11
-vi, 14
-wait, 9

