

# СУБД Firebird, утилита nbackup

Перевод [https://firebirdsql.org/file/documentation/reference\\_manuals/user\\_manuals/Firebird-nbackup.pdf](https://firebirdsql.org/file/documentation/reference_manuals/user_manuals/Firebird-nbackup.pdf)

Автор Paul Vinkenoog, последняя редакция Октябрь 2022, версия 1.7, Mark Rotteveel, Alexey Kovyazin.

Перевод на русский язык, исправления и дополнения: Кузьменко Д.В. (kdv), iBase.ru, август-сентябрь 2019. Обновление – 15.10.2024г, 05.02.2025г.

## Оглавление

СУБД Firebird, утилита nbackup.....	1
1. Введение .....	2
2. Обзор возможностей Nbackup .....	2
2.1 Преимущества nbackup .....	2
2.2 Недостатки nbackup.....	2
3. Функции и параметры.....	3
4. Создание резервных копий и восстановление .....	4
4.1 Полные бэкапы .....	5
4.2 Инкрементные бэкапы.....	7
4.3 Бэкапы баз данных на сырых устройствах (raw devices) .....	9
4.4 Отключение триггеров базы данных (Firebird 2.1+) .....	9
4.5 Кэширование ввода-вывода (Direct I/O) (Firebird 2.1.4+).....	9
4.6 Информационные параметры (Firebird 2.5+).....	10
4.7 Бэкапы по сети (Firebird 2.5+) .....	10
4.8 Практическое применение .....	11
4.9 Что дальше? .....	13
5. Блокирование и разблокирование .....	13
5.1 Блокирование базы данных и самостоятельное копирование.....	13
5.2 Восстановление бэкапа после “nbackup –L” .....	14
5.3 Как это работает .....	15
5.4 Блокирование баз на сырых устройствах .....	15
5.5 Блокирование и разблокирование базы при бэкапе виртуальной машины .....	16
6. Настройка дельта-файла .....	16
7. Лог истории бэкапов .....	17
8. Техническая информация .....	17

Изменения и дополнения в Firebird 3.0 .....	18
Изменения и дополнения в Firebird 4.0 .....	19
Изменения и дополнения в Firebird 5.0 .....	20
Appendix A: История документа .....	21
Appendix B: Замечания по лицензии.....	22

## 1. Введение

*Nbackup* – утилита резервного копирования, включенная в Firebird начиная с версии 2.0. Она предоставляет возможности, отсутствующие у *gbak* (штатной утилиты резервного копирования Firebird), но не заменяет *gbak*. Обе утилиты имеют свои достоинства и недостатки, и их вполне можно использовать совместно.

## 2. Обзор возможностей Nbackup

Nbackup позволяет выполнять два типа задач:

- Делать полные и инкрементные бэкапы. Инкрементные бэкапы содержат только изменения относительно предыдущего бэкапа.
- Блокировать базу данных для ее онлайн-копирования средствами операционной системы или другими утилитами. В этом режиме nbackup ничего не сохраняет и не копирует. Он просто создает условия, при которых копирование файла базы данных является безопасным.  
Блокирование базы является необходимым при бэкапах виртуальных машин – без блокирования полученная в копии VM база будет в состоянии как после нажатия кнопки Reset. См. [Блокирование и разблокирование базы при бэкапе виртуальной машины](#).

Обе этих задачи могут выполняться над активной базой данных, без необходимости отключения пользователей. Результирующий бэкап (или копия файла базы данных) будет содержать базу данных на момент начала бэкапа. Только SYSDBA и владелец базы данных (или администраторы операционной системы) могут делать бэкап, но при этом любой пользователь может восстановить базу данных из такого бэкапа. В этом плане nbackup не отличается от gbak.

### 2.1 Преимущества nbackup

- *В обоих режимах*: высокая скорость (насколько позволяет оборудование и операционная система), поскольку nbackup оперирует страницами БД, а не записями, и не проверяет данные на страницах.
- *В режиме бэкапа/восстановления*: экономия времени и дискового пространства, поскольку нет необходимости делать полный бэкап каждый раз. Это позволяет сильно упростить задачу резервного копирования много-гигабайтных баз данных.
- *В режиме блокировки*: полная свобода выбора способа копирования файла БД.

### 2.2 Недостатки nbackup

- Nbackup не будет чистить мусор, освобождать место в базе данных, сжимать бэкап или уменьшать базу при восстановлении из бэкапа, и так далее

- Вы не можете поменять владельца БД через резервное копирование и восстановление, как это может gbak
- Nbackup не делает “transportable backup”, то есть, полученная копия базы данных может быть несовместима с Firebird на другом аппаратном обеспечении (kdv: имеются в виду платформы с разным порядком байт в integer – big-endian и little-endian. К разным ОС на одной платформе это не имеет никакого отношения)
- В настоящий момент nbackup не работает с много-файловыми базами данных (kdv: которые являются атавизмом и сейчас уже не используются)
- Nbackup может оперировать только с локальными файлами баз данных. Nbackup можно запускать удаленно (через Services API и fbsvcmgr), но сами базы и их копии все равно должны быть локальными по отношению к серверу Firebird
- За исключением доступа через сервисы (Services API и fbsvcmgr) nbackup требует прямого доступа к файлу базы данных (в отличие от gbak).

### 3. Функции и параметры

В таблице перечислены параметры nbackup с общим описанием их назначения. Если столбец «Добавлено» пуст, то параметр существовал с момента появления nbackup в Firebird 2.0.

Параметр	Назначение	Добавлено
-B {n   guid} <database> [<имяфайла>]	Сделать бэкап уровня n в указанный файл, или, начиная с 4.0, сделать инкрементный бэкап с момента конкретного guid копии базы (см. <a href="#">далее</a> )	
-R <database> [<имяфайла> ...]	Восстановить базу данных из указанных файлов	
-L <database>	Заблокировать базу данных	
-UN[LOCK]	Разблокировать заблокированную базу данных	3.0
-N <database>	Разблокировать заблокированную базу данных	
-F[IXUP] <database>	Разблокировать восстановленную базу данных	
-S	Показать размер базы данных в страницах после блокирования (совместно с -L)	2.1
-T	Игнорировать триггеры уровня базы данных (совместно с -B, -L, -N), с 3.0 рекомендуется использовать nodbtriggers	2.1
-NODBTRIGGERS	игнорировать триггеры БД (при -B, -L или -U)	3.0
-D on off	Прямой ввод-вывод включить (on) или выключить (off) (совместно с -B)	2.1.4
-I[NPLACE]	восстановить инкрементный бэкап в существующую базу (может повредить БД при некорректном использовании); требуется -RESTORE. (см. <a href="#">далее</a> )	4.0
-DE[COMPRESS] команда	Для извлечения из архива, требуется -RESTORE	3.0
-SEQ[UENCE]	Сохранить номер сегментов репликации, нужен для -RESTORE или -F на сервере репликации	4.0
-U <username>	Имя пользователя (совместно с -B, -L, -N)	
-P <password>	Пароль пользователя (совместно с -B, -L, -N)	
-RO[LE]	указать имя роли (совместно с -B, -L, -N)	3.0
-FE <имяфайла>	Взять пароль из файла (совместно с -B, -L, -N)	2.5
-CLEAN_HIST[ORY]	Очистить старые записи в истории бэкапов (см. <a href="#">далее</a> ). Используется только с -KEEP	5.0

-K[EEP] <n> R[OWS]   D[AYS]	Сколько строк или дней (от сегодня) нужно сохранить в истории бэкапов. Используется только вместе с –CLEAN_HIST	5.0
-Z	Вывести версию (отдельно или совместно с -B, -R, -L, -N, -F)	2.5
-?	Вывод справки по всем параметрам	2.5

В зависимости от выбранной основной функции (-B, -L, -N, -R или -F) nbackup может требовать определенный уровень доступа к базе данных: соединение с сервером Firebird, прямой доступ к файлам, или и то и другое. Следующая таблица показывает детали:

Параметр	Назначение	Доступ
-B	Бэкап	сервер и файл
-R	Восстановление	файл
-L	Блокирование	сервер
-UNLOCK	Разблокирование (после –L)	сервер
-N	Разблокирование (после –L)	сервер
-F	Разблокирование после восстановления (-R)	файл

*Замечание (kdv):* регистр параметров не имеет значения, может быть указано –B или –b, и так далее. Параметры приведены в верхнем регистре как минимум потому, что один из параметров – L в нижнем регистре –l выглядит как -1 в большинстве шрифтов.

Когда требуется доступ к серверу (с параметрами –B, -L и –N/-UN), пользователь должен указать имя пользователя и пароль (параметрами –U и –P | –FE, или через переменные пользователя ISC\_USER и ISC\_PASSWORD), или быть root на Unix или доверенным пользователем на Windows. Когда требуется доступ к файловой системе (с –B, -R и –F), пользователь должен иметь достаточные привилегии к файлу базы данных для чтения и записи. Когда требуется исключительный доступ к файлу (с –R и –F), пользователю не нужен логин к Firebird, и сервер Firebird может быть не запущен или отсутствовать.

Для параметров –F и –R начиная с 4.0 есть дополнительный параметр –SEQ[UENCE] – он нужен для сохранения GUID базы данных реплики и номера сегментов репликации, эквивалентных оригинальной базе данных (мастер-базе). В этом случае репликация (на реплике) продолжится с того момента, когда был сделан физический бэкап на мастере (nbackup –b 0).

Например:

```
nbackup -R[ESTORE] <database file> <backup file> -SEQ[UENCE]
```

```
nbackup -F[IXUP] <database file> -SEQ[UENCE]
```

Замечание: В приведенных таблицах и тексте имеется в виду доступ к базе данных. Доступ к файлам бэкапов – всегда на уровне файловой системы.

#### 4. Создание резервных копий и восстановление

Для начала – утилита nbackup находится там же, где и другие утилиты Firebird (gbak, gstat...).

Типичное расположение для Firebird версий 1-2.5 это каталог bin установки, для 3.0 – корневой каталог установки.

Как и другие утилиты Firebird, nbackup не имеет графического интерфейса. Вы запускаете его из командной строки, или из командного файла, или из приложения.

Важно: На некоторых системах с высокой нагрузкой nbackup версии 2.0.3 и ниже мог вызвать проблемы, приводящие к блокировкам, или даже повреждению базы данных. Поскольку эта проблема редкая, вполне достаточно обновиться до Firebird 2.0.4 или выше, если вы хотите комфортно использовать nbackup.

Если иметь в виду большие базы данных на Linux, использование прямого ввода-вывода (без использования файлового кэша ОС) может давать разные результаты по скорости. См.

[Кэширование ввода-вывода \(Direct I/O\)](#)

## 4.1 Полные бэкапы

### 4.1.1 Как делать полный бэкап

Чтобы сделать полный бэкап базы данных, выполните команду

```
nbackup -B 0 <database> [<имяфайла>] -U <user> - P <password>
```

Например

```
C:\Data>nbackup -B 0 inventory.fdb inventory_2019-Aug-01.nbk -u SYSDBA -p masterkey
```

Замечание (kdv): при использовании дат и времени в именах бэкапов используйте формат «год-месяц-день» (или японский формат уууutmdd) – так вы получите корректную сортировку имен файлов по порядку. Не используйте формат примера оригинального документа – там inventory\_1-Mar-2006.nbk, сортировка имен в папке в этом случае работать не будет (т.е. будет, но порядок будет не по дате).

Или, для Firebird 2.5 и выше, можно указать нестандартный порт (если используется)

```
C:\Data>nbackup -B 0 localhost/3051:c:\data\inventory.fdb inventory_2019-Aug-01.nbk -u SYSDBA -p masterkey
```

В Firebird 3.0 и выше, в случае успешного завершения бэкапа, nbackup выдаст статистику

```
time elapsed 1 sec
```

```
page reads 307
```

```
page writes 307
```

(разумеется, для ваших файлов числа будут другие)

- Параметр –B означает «бэкап». Уровень бэкапа 0 производит полный бэкап, т.е. полную копию базы данных. Уровни бэкапа выше 0 используются для инкрементных бэкапов, которые описаны дальше.
- Вместо имени базы данных можно указать ее алиас
- Вместо имени файла бэкапа можно указать stdout. В этом случае вывод можно перенаправить, например, в архиватор.
- Параметры –U и –P можно не указывать при следующих условиях:
  - Установлены корректные значения в переменных среды ISC\_USER и ISC\_PASSWORD, это SYSDBA или владелец БД и пароль
  - Вы вошли на Linux как root, это эквивалентно SYSDBA
  - На Windows: В firebird.conf включена доверенная аутентификация (trusted authentication), и вы вошли в Windows с правами администратора, которому доступен файл базы данных. Для Firebird 2.1 это дает вам права SYSDBA. Для Firebird 2.5 и выше должно быть установлено AUTO ADMIN MAPPING.

Для упрощения, в дальнейшем параметры –U и –P не указываются в примерах

- Начиная с Firebird 2.5 вместо –P <пароль> можно указывать –FE <имяфайла>. В этом случае nbackup прочитает пароль из указанного файла. В случае –FE пароль не указывается в командной строке, и это позволяет исключить доступ к паролю людей, которые могут посмотреть историю команд в командной строке (например, командой w на Unix или в скрипте)
- В Firebird 2.1 и выше параметр –T отключает триггеры базы данных. Подробнее см. раздел «[Отключение триггеров БД](#)»
- В Firebird 2.1.4 и выше возможно отключить кэширование БД файловой системой (включить прямой ввод-вывод) параметром –D on (или выключить прямой ввод-вывод –D off). Подробности см. далее в [соответствующем разделе](#).
- Разные параметры (–B, –U и др.) могут быть указаны в любом порядке. Разумеется, за каждым параметром должен следовать его аргумент. В случае –B это три аргумента – уровень бэкапа, имя базы данных, и имя файла бэкапа – именно в таком порядке.
- Если параметр –B указан последним, вы можете не указывать имя файла бэкапа. В этом случае nbackup сам сгенерирует имя вида database.fdb-0-20190807-2105.nbk, где 0 это уровень, 20190807 – дата, 2105 – время. Но это может привести к ошибке, если бэкап одного и того же уровня делается два раза в одну и ту же минуту  
! (kdv): вопреки оригиналу это не работает в 3.0, 4.0 и 5.0. Выдается ошибка  
*Missing parameter for switch -b*

Важно: Не используйте nbackup на много-файловых базах данных. Это может привести к потере данных (неполный и поврежденный бэкап), при том что nbackup не выдаст сообщение об ошибке.

### Немного о том, как это устроено

Замечание: Это необязательная информация. Но она дает вам чуть больше представления о том, что происходит при выполнении nbackup –B

1. Сначала база данных блокируется (выставляется внутренний флаг состояния). С этого момента любые изменения базы данных записываются во временный файл – так называемый «дельта-файл»
2. Производится бэкап. При –b 0 это эквивалент копирования файла, однако для приведения копии в рабочее состояние над ней нужно выполнить nbackup –F. При –b n, где n больше 0 – производится инкремент, который можно восстановить (вместе с полной копией) только командой –R.
3. По завершении бэкапа, содержимое дельта-файла записывается в базу данных. База данных разблокируется, и дельта-файл удаляется.

Шаги 1 и 3 обеспечиваются новыми командами SQL – ALTER DATABASE BEGIN BACKUP и ALTER DATABASE END BACKUP. Эти команды не выполняют сам бэкап (копирование), они только создают условия, при которых база данных может быть успешно скопирована. Выполнять эти операторы самостоятельно нет необходимости (т.к. из SQL вы не имеете доступа к файлам), nbackup сделает всё за вас.

#### 4.1.2 Восстановление полных бэкапов

Полный бэкап восстанавливается следующей командой

```
nbackup -R <database> [<файлбэкапа>]
```

Например

```
C:\Data>nbackup -R inventory.fdb inventory_2019_08_07.nbk
```

- Не нужно указывать «уровень» восстановления
- При восстановлении параметр `-R` должен быть указан последним. Причина этого будет объяснена далее.
- Вместо имени базы данных можно указать алиас
- Если указано имя существующей базы данных, то будет выдана ошибка
- Здесь также можно не указывать имя бэкапа. В этом случае `nbackup` запросит имя файла (Внимание: в Firebird 2.0 эта функциональность не работает. Исправлено в 2.0.1)
- Восстановление полностью работает на уровне файловой системы, и может быть выполнено без работающего Firebird. Любые значения, указанные для параметров `-U` и `-P` игнорируются, поэтому указывать эти параметры не нужно. То же самое в отношении опции `-FE`, однако `nbackup` все равно будет пытаться прочитать пароль из файла, и если произошла ошибка, вся операция будет отменена.

## 4.2 Инкрементные бэкапы

### 4.2.1 Как сделать инкрементный бэкап

Чтобы сделать инкрементный (разностный) бэкап нужно указать уровень бэкапа больше 0. Инкрементный бэкап уровня 1 и выше всегда содержит только те изменения базы данных, которые произошли с момента предыдущего бэкапа уровня N-1.

Примеры:

Через день после полного бэкапа (уровня 0) вы делаете один уровня 1:

```
C:\Data>nbackup -B 1 inventory.fdb inventory_20190808_1.nbk
```

**Важно!** Рекомендуется в имени инкрементных бэкапов указывать уровень такого бэкапа, чтобы не запутаться. Также, для сортировки имен бэкапов удобно использовать формат даты `yyyymmdd`.

Этот бэкап будет содержать только те изменения, которые произошли в БД за последний день.

Еще через день, выполняете еще один бэкап уровня 1

```
C:\Data>nbackup -B 1 inventory.fdb inventory_20190809_1.nbk
```

Этот инкремент будет содержать изменения за 2 последних дня, с момента полного бэкапа (уровня 0), а не только изменения, которые произошли с момента предыдущего бэкапа уровня 1.

Замечание: Любой инкрементный бэкап должен успешно завершиться перед выполнением нового инкрементного бэкапа. В противном случае будет выдано сообщение «Database is already in the physical backup mode»

Через несколько часов сделаем бэкап уровня 2:

```
C:\Data>nbackup -B 2 inventory.fdb inventory_20190809_2.nbk
```



Этот свежий инкремент содержит изменения, начиная с самого последнего (свежего) инкремента уровня 1 (а не с бэкапа уровня 0).

*Замечание:* Все комментарии по поводу полного бэкапа относятся и к инкрементным бэкапам.

#### 4.2.2 Важное замечание по производительности

До Firebird 3.0 все инкрементные бэкапы (уровня 1, 2 и далее) читали всю базу данных каждый раз, для поиска измененных страниц. Это приводило к плохой производительности для баз 100 гигабайт и выше. Начиная с версии 3.0 инкрементные бэкапы читают только измененные страницы БД, без сканирования всего файла базы данных. Так что инкременты уровня 1, 2 и далее выполняются достаточно быстро, и оказывают малое влияние на производительность БД.

#### 4.2.3 Восстановление инкрементных бэкапов

При восстановлении базы данных из инкрементных бэкапов необходимо указать полную цепочку файлов, начиная от уровня 0 до требуемого уровня. База данных будет построена заново, шаг за шагом.

Синтаксис

```
nbackup -R <database> [<бэкап0> [<бэкап1> [...]]
```

Таким образом, восстановление из бэкапа уровня 2 по предыдущим примерам будет таким:

```
C:\Data\nbackup -R inventory.fdb inventory_2019_08_07.nbk  
inventory_20190808_1.nbk inventory_20190809_2.nbk
```

Разумеется, строка здесь «разделена» на две части только из-за форматирования. На самом деле это одна строка, которая завершается Enter в конце.

Комментарии (в дополнение к [восстановлению полного бэкапа](#)):

- Поскольку заранее неизвестно, сколько имен файлов вы укажете после параметра –R, nbackup считает все аргументы после –R именами файлов. Поэтому никакие другие параметры не должны следовать за –R.
- Нет лимита для номеров инкрементных бэкапов, но на практике редко имеет смысл заходить дальше уровней 3 или 4.

Что произойдет, если вы пропустите нужный файл в аргументах –R, или укажете некорректную цепочку файлов? В соответствии с примерами, допустим, укажете бэкап 0, затем первый бэкап уровня 1, и затем бэкап уровня 2, который построен на втором бэкапе уровня 1?

Бэкапов уровня 1 две штуки, поэтому, казалось бы, есть нормальная последовательность «0, 1, 2» файлов для восстановления БД.

К счастью, такая ошибка никогда не приведет к восстановлению некорректной БД. Каждый бэкап (любого уровня) имеет свой собственный идентификатор. И, например, каждый бэкап уровня 1 (и выше) содержит идентификатор бэкапа уровня 0, на котором он базируется. При восстановлении nbackup проверяет эти идентификаторы. Если в цепочке «родительский» идентификатор «не тот», то операция отменяется и вы увидите сообщение об ошибке.



Замечание (kdv): Фактически, каждый инкрементный бэкап имеет «подмножество» инкрементов более высокого номера, из которых можно «склеивать» результирующую базу. Например, инкремент уровня 1 можно «склеивать» только с теми инкрементами уровня 2, которые были сделаны до момента нового инкремента уровня 1.

#### 4.3 Бэкапы баз данных на сырых устройствах (raw devices)

Базы Firebird не обязательно должны быть файлами, они могут располагаться на так называемых сырых (raw) устройствах, не имеющих файловой системы. Здесь может возникнуть вопрос, где будет в этом случае располагаться дельта-файл. Например, на Posix-системах, если база данных находится в /dev/hdb5, то дельта при старте nbackup может получить имя /dev/hdb5.delta. Понятно, что такая ситуация нежелательна.

Начиная с Firebird 2.1 nbackup отказывается работать с базой на сырых устройствах до тех пор, пока расположение дельта-файла не задано явно. Эта тема обсуждается в разделе [«Настройка дельта-файла»](#).

#### 4.4 Отключение триггеров базы данных (Firebird 2.1+)

В Firebird 2.1 появились триггеры уровня базы данных. Некоторые типы этих триггеров могут срабатывать при подключении к базе данных и отключении от нее. При бэкапе nbackup подключается к базе данных обычным способом (в некоторых версиях больше чем одним коннектом). Для предотвращения срабатывания триггеров уровня базы данных введен новый параметр –Т. Помните, что аналогичный параметр у gbak и isql называется –nodbtriggers. Соответственно, в 3.0 появился аналогичный параметр nbackup - -NODBTRIGGERS, а использовать параметр –Т больше не рекомендуется.

#### 4.5 Кэширование ввода-вывода (Direct I/O) (Firebird 2.1.4+)

Первоначально nbackup использовал прямой ввод-вывод только под Windows NT (и последующих 2000, 2003 и так далее). На других операционных системах прямой ввод-вывод был выключен. Это приводило к проблемам на некоторых Linux, поэтому в версиях 2.0.6 и 2.1.3 прямой ввод-вывод также был включен по умолчанию на Linux. Однако, это создало определенные проблемы на некоторых конфигурациях Linux. В 2.1.4 и 2.5 оригинальное поведение было восстановлено, но в этот раз включить или выключить прямой ввод-вывод можно параметром –D:

```
nbackup -B 0 cups.fdb cups.nbk -D on -- прямой ввод-вывод включен
nbackup -B 0 mugs.fdb mugs.nbk -D off -- прямой ввод-вывод выключен
```

Также как и параметр, аргументы on и off являются нечувствительными к регистру.

Прямой ввод-вывод относится только к бэкапу, не к восстановлению. На Windows –D on реализуется использованием FILE\_FLAG\_NO\_BUFFERING. На других системах это O\_DIRECT и POSIX\_FADV\_NOREUSE.

Последние два на некоторых системах могут не работать. В этом случае они (или один из них) молча не используется. Даже если пользователь указал –D on явно, это не выдаст на таких системах предупреждения или сообщения об ошибке.

**Примечание kdv:** Запутали с этими чередованиями умолчаний неимоверно, поэтому рекомендую всегда явно указывать эту опцию (-D on или –D off), в этом случае не ошибетесь.

**Важно! (kdv):** «Прямой ввод-вывод» или Direct IO это ОТКЛЮЧЕНИЕ кэша файловой системы. То есть, -D on отключает кэш файловой системы, а -D off – включает.

Тут нужно заметить, что по умолчанию, если в firebird.conf значение параметра DBDefaultCachePages меньше FileSystemCacheThreshold, то файловый кэш для БД включен (на любой ОС), и это сильно помогает производительности в архитектурах Classic и SuperClassic (да и Superserver тоже).

Однако, если на таких системах выполнить бэкап nbackup с опцией -D on, это немедленно ВЫКЛЮЧИТ файловый кэш для базы данных, и производительность может резко упасть.

Восстановится она только после окончания бэкапа, пока nbackup не «отпустит» базу данных и кэширование базы файловой системой опять не заработает.

Наоборот, если вы используете SuperServer, и отключили файловый кэш у БД в firebird.conf или databases.conf, включение файлового кэша nbackup -D off может привести к выпадению процессов в виртуальную память, т.к. операционной системе потребуется память для размещения там файлового кэша базы данных.

Таким образом, в обычной ситуации следует всегда явно использовать -D off, а когда вы точно знаете, что кэш БД выключен в firebird.conf - -D on.

! На некоторых устройствах с размером сектора 1024 байт опция -D ON может приводить к ошибке

*IO error reading file, the parameter is incorrect*

<https://github.com/FirebirdSQL/firebird/issues/7873>

Исправлено в версиях 4.0.5, 5.0.1. (3.0 – неясно, актуальная 3.0.12).

#### 4.6 Информационные параметры (Firebird 2.5+)

Кроме упомянутых ранее -FE и -D nbackup в Firebird 2.5 дополнительно имеет два параметра:

-Z показывает информацию о версии. Этот параметр может быть использован независимо, или вместе с другими параметрами, -B, -R, -L и т.д.

-? показывает все возможные параметры командной строки. Если указывается -?, то все остальные параметры игнорируются.

#### 4.7 Бэкапы по сети (Firebird 2.5+)

nbackup работает с локальными базами данных. Но в Firebird 2.5 и выше, операции nbackup могут быть выполнены удаленно через Services Manager. Для этого используется утилита *fbsvcmgr*. Она находится в той же папке, что и nbackup. Первый аргумент всегда “hostname:service\_mgr”, где hostname это имя удаленного сервера (где работает Firebird). Другие параметры:

- user username
- password password
- action\_nbak
- action\_nrest
- nbk\_level n
- dbname database
- nbk\_file filename
- nbk\_no\_triggers
- nbk\_direct on|off

Выполнить nbackup на машине srv можно следующим образом:

```
fbsvcmgr srv:service_mgr -user SYSDBA -password masterkey, -  
action_nbak -nbk_level 0 -dbname c:\databases\countries.fdb -nbk_file  
c:\databases\countries.nbk
```

Или инкрементный бэкап

```
fbsvcmgr srv:service_mgr -user SYSDBA -password masterkey, -  
action_nbak -nbk_level 1 -dbname c:\databases\countries.fdb -nbk_file  
c:\databases\countries_1.nbk
```

Для восстановления базы данных

```
fbsvcmgr srv:service_mgr -user SYSDBA -password masterkey, -  
action_nrest -dbname c:\databases\countries_restored.fdb -nbk_file  
c:\databases\countries.nbk -nbk_file c:\databases\countries_1.nbk
```

Каждая такая строка должна быть введена как единое целое, без переводов строки.

Комментарии:

- Services Manager всегда требует указания имени пользователя и пароля, явного при помощи опций `-user` и `-password`, или автоматического (`root` или `trusted authentication`). Переменные `isc_user` и `isc_password` не используются. `AUTO ADMIN MAPPING` не действует при удаленном коннекте.

**Примечание kdv:** пропустил перевод оригинала на тему `trusted authentication`, неинтересно, см. оригинальный документ.

- Восстановление (`-action_nrest`) также требует аутентификации, но она реально не используется. Поэтому не нужно быть владельцем БД, `SYSDBA` или суперпользователем.

**Примечание kdv:** дальше в оригинальном документе идет описание какого-то потенциального риска безопасности, которое я не стал переводить. `nbackup` никак не меняет `owner` базы данных, и кто этот файл восстановил из бэкапа, не имеет значения. Кому интересно, читайте это в оригинальном документе.

- Services Manager может быть использован локально. В этом случае первый аргумент становится `service_mgr`, без имени сервера. При локальном коннекте `AUTO ADMIN MAPPING` действует, как будто вы указали `localhost` в качестве имени сервера. Локальное использование Services Manager имеет смысл если вы не имеете файлового доступа к базе данных и каталогам с бэкапами, а Firebird – имеет. Если же у вас есть соответствующие права, то имеет смысл использовать непосредственно `nbackup`, а не `fbsvcmgr`.
- Указание `-nbk_no_triggers` или `-nbk_direct` с командой `-action_nrest` приведет к сообщению об ошибке. Сам `nbackup` будет просто игнорировать параметры `-T/-NODBTRIGGERS` и `-D` при указании команды `-R`.
- Вместо имени файла базы данных вы можете использовать алиас.
- Путь к базе данных (включая имя) ограничен 255 символами.

## 4.8 Практическое применение

Обычная схема использования `nbackup` может выглядеть так

- В начале каждого месяца выполняется полный бэкап уровня 0
- каждую неделю – уровень 1
- каждый день – уровень 2

- каждый час – уровень 3

Если вы сохраняете эти бэкапы, то можете восстановить базу данных на любой час любого дня месяца. В этом случае используется максимум 4 уровня (от 0 до 3). Вы можете использовать другие интервалы времени, в зависимости от загрузки сервера, например, уровни 0 и 1 можно делать на выходных, и уровень 2 ночью.

Если вы не хотите хранить все бэкапы вечно, можно использовать схему:

- бэкапы уровня 3 удаляются каждый 8й день
- файлы уровня 2 – каждый месяц
- файлы уровня 1 – каждые 6 месяцев
- полный бэкап оставляем на 2 года, затем удаляем

Это просто примеры, выбирайте самостоятельно частоту бэкапов и необходимые уровни.

**Примечание *kdv*:** вы должны заранее оценить скорость вашей дисковой подсистемы и объемы инкрементов, чтобы бэкап самого последнего используемого вами уровня успел выполниться в назначенный интервал времени. То есть, если бэкап уровня 3 делается каждый час, то он сам по себе не должен длиться больше часа ни при каких условиях (загруженности системы и скорости дисков).

**Примечание *kdv*:**

`nbackup` уровня 1 позволяет вам понять, какое количество изменений (страниц) происходит в вашей базе за некий интервал времени. Если вы делаете `nbackup -b 1` через день после `nbackup -b 0`, то размер файла уровня 1 будет как раз содержать количество измененных страниц за 1 день.

Это не эквивалентно количеству изменений за день вообще, т.к. одна страница БД может изменяться множество раз, но в бэкап уровня 1 попадет только самое последнее изменение этой страницы.

Одной из неожиданных для меня схем в одной из компаний был отказ от бэкапов уровня выше 1. То есть, после бэкапа уровня 0 весь месяц, каждый день, делаются бэкапы только уровня 1. Понятно, что каждый день файл бэкапа уровня 1 может расти, но опять же, это зависит от количества изменений разных страниц, а не одних и тех же страниц.

Для примера (конкретная база данных, её размер и размер инкрементов), база 10 гигабайт. `nbackup` уровня 0 делается 1 раз в месяц. Бэкап уровня 1 делается раз в неделю, и занимает 1 гигабайт, бэкапы уровня 2 каждый день по 800 мегабайт, бэкапы уровня 3 раз в час по 300 мегабайт. В итоге, если хранить все это за месяц, получается 136 гигабайт для 10гб базы данных.

Обратите внимание, что бэкапы уровня 3, которые делаются 1 раз в час, занимают по 300мб каждый, а бэкап уровня 2, который делается 1 раз в день, занимает 800мб. Если умножить 300мб на 24 часа, мы получим необходимость хранения 7.2 гигабайт ежечасных бэкапов вместо одного 800 мегабайтного за день. Но это цена возможности восстановления состояния базы данных на каждый час, а не на каждый день.

Поэтому, при планировании бэкапов необходимо учитывать стоимость возможности восстановления на каждый час и стоимость хранения этих данных.

Другой пример – база данных 375 гигабайт, бэкап уровня 0 длится 45 минут, бэкапы уровня 1 (каждый день) длятся не более 20 минут, и занимают до 25 гигабайт. В результате, объем файлов бэкапа уровня 0 плюс бэкапы уровня 1 за 30 дней занимают 1.1 терабайт.

Кстати, просто для сравнения. Восстановление из nbackup такой БД (375 гигабайт) занимает 1 час, а восстановление из бэкапа gbak – 17 часов. Разумеется, «часы» здесь приведены для конкретной дисковой подсистемы. Поэтому вам нужно самостоятельно проверить, сколько времени у вас занимает полный бэкап, инкременты, и восстановление БД.

**Внимание!** При работе с nbackup необходимо следить за двумя параметрами – как долго существует дельта-файл, и какой размер он имеет. Критическими являются существование дельта-файла больше двух-трех часов, или если размер дельта-файла превысил более трети размера БД. Оба этих критических значения указывают на то, что разблокирование БД после старта nbackup не произошло по какой-то причине, и все изменения до сих пор пишутся в дельта-файл, а не в базу данных.

Посмотреть состояние базы данных (и состояние бэкапа) можно командой `gstat -h <база данных>`.

#### 4.9 Что дальше?

К этому моменту вы уже знаете всё о бэкапах и восстановлении. Далее приведена информация о специфических режимах работы nbackup, которые могут быть вам интересны.

### 5. Блокирование и разблокирование

Если вы хотите использовать собственные средства бэкапа, или просто сделать файловую копию БД, то режим блокирования и разблокирования БД будет интересен для вас. «Блокирование» означает что база данных будет «заморожена» на некоторое время, и в ней не будут производиться никакие изменения. Изменения будут записаны в дельта-файл, и при разблокировании дельта-файл будет записан обратно в базу данных.

#### 3.5 Блокирование базы данных и самостоятельное копирование

Типичный сеанс самостоятельного бэкапа выглядит так:

1. Блокируем базу данных параметром `-L`  
`nbackup [-U user -P password] -L database`
2. Теперь можно скопировать базу данных любыми утилитами (`copy /b`, `xcopy`, и т.д...)
3. Разблокируем базу данных параметром `-N` или параметром `-UN` (с Firebird 3.0)  
`nbackup [-U user -P password] -N database`

На шаге 3 все изменения, которые происходили с базой данных во время шага 2, будут записаны в БД из дельта-файла, а сам дельта-файл после этого будет удалён.

Результирующая копия базы данных будет содержать всё, что было в базе на момент блокировки.

Комментарии:

- Вы можете указать алиас вместо имени БД
- параметры `-U` и `-P` могут не указываться, если вы установили переменные `ISC_USER` и `ISC_PASSWORD`, или являетесь `root`, или доверенным пользователем на Windows.
- Начиная с Firebird 2.5 вы можете вместо `-P <пароль>` указать `-FE <имяфайла>`

- Обе опции, -L и -N/-UN, производят обычный коннект к базе данных, поэтому в Firebird 2.1 и выше может быть имеет смысл дополнительно указать -T/-NODBTRIGGERS ([Отключение триггеров БД](#)).
- Если вы делаете бэкап базы данных, расположенной на «[сыром устройстве](#)», то вероятно, вам не помешает параметр -S для получения информации о базе данных.
- Можно указать параметр -Z, который выводит информацию о версии.

Примечание: Что относится к бэкапу и восстановлению, точно так же относится и к блокированию-разблокированию многофайловых баз данных. Не используйте nbackup на многофайловых базах данных, совсем!

## 5.2 Восстановление бэкапа после “nbackup -L”

Скопированная в момент блокирования база данных также является заблокированной, поэтому вы не можете её использовать сразу после копирования (у нее отсутствует дельта-файл, а скопировать дельта-файл можно только остановив службу Firebird до разблокирования основной БД). Корректный сценарий использования копии выглядит так:

1. копируем заблокированную БД нужными нам средствами
2. Разблокируем скопированную БД, но не параметром -N/-UN, а параметром -F (fixup)  
nbackup -F <database>

Здесь можно использовать алиас вместо имени файла базы данных, и указывать параметр -Z. Указание других параметров не имеет смысла.

Почему есть два параметра разблокирования, -N и -F?

- -N вначале проверяет, были-ли изменения БД с момента ее блокирования -L. Если да, то дельта-файл применяется к БД, после чего БД разблокируется, а дельта-файл удаляется.
- -F просто меняет флаг состояния БД из «заблокировано» в «нормальный».

! При изготовлении копии базы для репликации при помощи nbackup -b 0 необходимо эту копию разблокировать с опцией -seq:

```
nbackup dbreplica -f -seq
```

**Примечание *kdv*:** в некоторых версиях Firebird были случаи, когда невозможно сделать nbackup -N/UN, выдавалось сообщение об ошибке, что БД всё равно осталась в состоянии backup lock. Иногда причиной ошибки было наличие дельта-файла без записи в rdb\$backup\_history (см. [далее](#)), или наоборот – дельта-файла нет, а запись в rdb\$backup\_history есть. В этом случае необходимо подключиться к БД isql (или другим инструментом), и выдать команду ALTER DATABASE END BACKUP.

**Примечание *kdv*:** опцию -F можно использовать вместо -N если в процессе nbackup (от момента блокирования БД) произошло повреждение дельта-файла. Разумеется, в этом случае все изменения, которые были в дельта-файле, пропадут, т.е. не будут записаны в БД.

! не ошибитесь! Не указывайте -F после -L, если дельта-файл цел и вы не хотите потерять все изменения в базе с момента -L.

! опция -F работает в том числе, если файл базы данных открыт каким-нибудь коннектом. Именно поэтому есть опасность в «отмене дельты» на рабочей БД. Есть предложение, что опция -F не

должна работать на БД, открытой сервером:

<https://github.com/FirebirdSQL/firebird/issues/8226>

**Примечание kdv:** некоторые администраторы нашли способ безопасного обновления структуры базы данных (наката скриптов, и проч.). Перед обновлением они делают –L, затем запускают программу обновления БД или выполняют скрипты обновления. Затем, если всё прошло нормально – делают –N/-UN. Если нет – делают –F, таким образом «отменяя» все изменения, которые попали в дельта-файл (после чего нужно будет удалить файл дельты вручную – да и применить его к БД будет уже никак нельзя). Понятно, что в этом случае пропадут и все изменения данных, которые были сделаны при активной дельте.

Итак, вы используете

- –N/-UN после того как сделали копию файла самостоятельно (после –L)
- –F если хотите сделать рабочей БД из скопированного файла.

### 5.3 Как это работает

Этот раздел не содержит обязательных знаний, но дает дополнительную информацию для понимания работы некоторых параметров nbackup.

**nbackup –L** делает следующее:

1. подсоединяется к БД
2. стартует транзакцию
3. выполняет ALTER DATABASE BEGIN BACKUP (этот оператор уже обсуждался в разделе [«Восстановление полных бэкапов»](#))
4. делает коммит транзакции
5. отсоединяется от базы данных

**nbackup –N/-UN** выполняет те же самые шаги, только на шаге 3 «...END BACKUP».

**nbackup –F** работает так:

1. открывает файл базы данных
2. внутри файла меняет флаг состояния nback\_state\_stalled на nback\_state\_normal.
3. закрывает файл базы данных

nbackup –F работает только с файлом, поэтому может быть выполнен без работающего сервера Firebird. Любые аргументы параметров –U, –P, или –FE игнорируются, так же как и для nbackup –R.

### 5.4 Блокирование баз на сырых устройствах

Как уже было упомянуто в разделе [«Бэкапы баз данных на сырых устройствах»](#), проблемы могут возникнуть если дельта-файл создается для базы данных, которая находится на сыром устройстве (raw device). Поэтому, в Firebird 2.1 и выше nbackup отказывается работать с базами на сырых устройствах до тех пор, пока не будет задано явное расположение дельта-файла. Как это сделать, написано в [следующем разделе](#).

Есть и другая проблема, если вы блокируете и пытаетесь скопировать сырое устройство – вы не знаете реального размера базы данных! Само устройство может иметь размер 10 гигабайт, а база данных на нем может занимать всего 200 мегабайт. Чтобы не копировать всё устройство целиком, и не тратить зря время и место, в Firebird 2.1 введен новый параметр nbackup –S. Этот параметр



работает только вместе с параметром `-L`, и если указан, `nbackup` выведет размер базы данных в страницах в `stdout`, после того как БД будет заблокирована. Поскольку размер приведен в количестве страниц, его нужно умножить на размер страницы, чтобы получить размер базы данных. Или, если вы пользуетесь утилитой `dd`, можно указать размер блока параметром `ibs`, равному размеру страницы.

## 5.5 Блокирование и разблокирование базы при бэкапе виртуальной машины

Использование средств копирования виртуальных машин без специальной подготовки базы данных может привести к поврежденной копии базы данных.

Firebird использует собственный кэш в памяти для ускорения операций, и использует сложные алгоритмы для обеспечения целостного состояния базы данных в каждый момент времени.

Однако, средства копирования виртуальных машин ничего не знают о кэше Firebird, и часто даже не рассматривают базы данных как файл случайного доступа.

В результате, после того, как бэкап виртуальной машины сделан, состояние этого бэкапа будет как после ресета виртуальной машины, и часто такая копия не будет целостной (т.е. будет считаться поврежденной). Шансы такого тем выше, чем больше пользователей меняют данные в базе данных, или если в этот момент работает `sweep`.

Такая ситуация может возникнуть в любом типе виртуальной машины, включая облачные системы.

Чтобы получить корректную копию базы Firebird в момент бэкапа виртуальной машины нужно перед этим заблокировать файл базы данных, и разблокировать по окончании бэкапа VM. Обычно средства бэкапа виртуальных машин позволяют выполнять пользовательские пре- и пост-скрипты, в которых вы можете указать блокирование и разблокирование базы Firebird.

## 6. Настройка дельта-файла

По умолчанию дельта-файл располагается рядом с базой данных. Имя файла такое же, как у базы данных, но к нему добавлено расширение `.delta`. Обычно такое поведение является приемлемым, однако иногда может потребоваться указать явно расположение дельта-файла, особенно если база данных находится на сыром устройстве (raw device). `nbackup` не имеет возможности указать расположение дельта-файла, это делается оператором SQL.

Подсоединитесь к базе данных, и выполните следующую команду:

```
alter database add difference file 'path-and-filename'
```

Такое расположение и имя дельта-файла будет в базе данных постоянным, оно хранится в системной таблице `RDB$FILES`. Для возврата к поведению по умолчанию используйте следующую команду:

```
alter database drop difference file
```

Указать путь к дельта-файлу можно и при создании базы данных:

```
create database 'path-and-dbname' difference file 'path-and-deltaname'
...
```

Замечания:

- Если вы укажете только имя файла в `[ADD] DIFFERENCE FILE`, без пути, то дельта-файл с этим именем будет создан не рядом с базой данных, а в «текущем каталоге» с точки

зрения сервера. На Windows это обычно системная папка (например, C:\Windows\system32). То же самое справедливо и для относительных имен каталогов.

- Указанный путь должен существовать. Firebird не будет создавать отсутствующие папки/каталоги.
- Если вы хотите поменять имя дельта-файла или путь к нему, сначала вы должны удалить (DROP) старое имя, и добавить (ADD) новое.

## 7. Лог истории бэкапов

Firebird сохраняет всю историю бэкапов nbackup в системной таблице rdb\$BACKUP\_HISTORY. Эта информация используется nbackup для собственных нужд, но также может использоваться для просмотра когда, какого уровня бэкап был сделан, и с каким именем файла.

Например, можно посмотреть информацию о 5ти последних бэкапах:

```
SELECT RDB$BACKUP_ID, RDB$TIMESTAMP, RDB$BACKUP_LEVEL, RDB$GUID,  
RDB$SCN, RDB$FILE_NAME  
FROM RDB$BACKUP_HISTORY  
ORDER BY RDB$TIMESTAMP DESC  
ROWS 5
```

Столбцы таблицы RDB\$BACKUP\_HISTORY:

Столбец	Описание
RDB\$BACKUP_ID	первичный ключ
RDB\$TIMESTAMP	дата и время бэкапа
RDB\$BACKUP_LEVEL	уровень бэкапа
RDB\$GUID	GUID бэкапа (используется для проверки взаимозависимостей разных уровней)
RDB\$SCN	
RDB\$FILE_NAME	имя файла бэкапа

Пояснения по столбцу RDB\$SCN вы можете прочитать в следующем разделе.

Содержимое таблицы RDB\$BACKUP\_HISTORY не сохраняется (и не восстанавливается) gbak-ом (потому что gbak при восстановлении создает базу данных заново, и эта база данных по содержимому страниц не имеет ничего общего с базой данных, бэкап которой был сделан gbak-ом).

В версии 5.0 появились дополнительные команды для очистки лога бэкапов (см. [далее](#)).

## 8. Техническая информация

nbackup делает бэкап копируя страницы базы данных, либо все (-B 0), либо те, которые были изменены с момента бэкапа предыдущего уровня (-B N). Например, бэкап уровня 1 содержит все страницы, которые были изменены с момента создания бэкапа уровня 0. Чтобы найти измененные страницы Firebird использует маркер, именуемый SCN (сокращение от System Change Number). Этот номер увеличивается каждый раз при изменении состояния бэкапа.

Кроме того, у базы данных есть три состояния

(находится в странице заголовка БД и может быть просмотрено gstat -h)

- nback\_state\_normal

- нормальное состояние БД. nbackup не работает, ввод-вывод не переадресуется, нет дельта-файла (для БД в нормальном состоянии информации о блокировке БД в выводе gstat нет)
- nback\_state\_stalled
  - бэкап в процессе. Ввод-вывод перенаправляется в дельта файл: все изменения записываются только в дельта-файл, измененные страницы читаются из дельта-файла, неизмененные страницы читаются из БД
- nback\_state\_merge
  - бэкап завершается. Изменения записываются из дельта файла в базу данных. Ввод-вывод всё еще перенаправлен: изменяемые страницы записываются как в БД, так и в дельта-файл, неизмененные страницы записываются из дельты в БД, измененные страницы читаются из дельты, неизмененные – из БД.

Эти состояния меняются из одного на другое, допустимы три варианта:

1. nback\_state\_normal (не бэкап) в nback\_state\_stalled (при начале бэкапа)
2. nback\_state\_stalled в nback\_state\_merge (при завершении бэкапа)
3. nback\_state\_merge в nback\_state\_normal (по окончании бэкапа)

SCN базы данных перед стартом бэкапа записывается в страницу заголовка БД, и в результате также попадает в бэкап. Самый первый бэкап получает SCN = 0, второй = 3, и так далее. Этот номер не зависит от уровня бэкапа. SCN используется для маркировки страниц в базе данных. Например, значения SCN на страницах БД:

0 – страница перед бэкапом

1 – страница, записанная в дельту или обновленная в ней во время бэкапа

2 – страница, модифицированная во время заливки дельты в базу данных

3 – страница, модифицированная после окончания бэкапа

Дальше, при очередном инкрементном бэкапе, сервер может определить, какие страницы были модифицированы после окончания предыдущего бэкапа, и какой это был уровень бэкапа.

Например, когда производится бэкап уровня 1, nbackup читает информацию о предыдущем бэкапе (уровня 0), и сохраняет в бэкап уровня 1 только те страницы, SCN которых больше, чем у бэкапа уровня 0. То же самое происходит с бэкапами уровня 2, и так далее.

При бэкапе и восстановлении с использованием gbak содержимое rdb\$backup\_history игнорируется, и также сбрасывается SCN для всех страниц (потому что restore делается в новый файл БД).

## Изменения и дополнения в Firebird 3.0

В Firebird 3 в базе данных хранится информация о SCN всех страниц, что позволяет серверу при очередном инкрементном бэкапе читать только измененные страницы. В предыдущих версиях Firebird для любого инкремента приходилось каждый раз сканировать всю базу данных.

Таким образом, если бэкап уровня 0 базы размером 31 гигабайт (конкретный пример) проходит за 6 минут, то бэкап уровня 1 при отсутствии изменений в базе с момента бэкапа уровня 0 будет выполнен за 14 секунд (прочитано 1984 страницы вместо чтения 3909536 страниц всей базы).

Дополнительно введен параметр `–decompress`, для распаковки сжатых бэкапов на линуксе при восстановлении БД из бэкапов. В качестве аргумента принимает командную строку, которая должна распаковать архив и передать в `stdout` распакованный файл бэкапа.

Примеры:

```
nbackup -de bzip2 -r e.fdb e.b0.bz2 e.b1.bz2
nbackup -de 'bzip2 -d -c' -r e.fdb e.b0.bz2 e.b1.bz2
nbackup -de 'bzip2 -d -c @"' -r e.fdb e.b0.bz2 e.b1.bz2
```

Все эти 3 команды приведут к одному и тому же результату – архивы будут распакованы и из них будет создана база данных `e.fdb`.

**Важно!** У Firebird 3 есть другая особенность – это флаг `swept` у страниц данных. Этот флаг устанавливается в `false`, как только на странице появляются версии записей, и устанавливается в `true`, когда sweeper или сборщик мусора полностью очищают страницу данных от мусорных версий. Главным здесь является то, что после `restore`, несмотря на отсутствие версий на страницах данных, все страницы данных помечены как `swept=false`. Это означает, что при очередном запуске sweeper все страницы данных могут поменяться (флаг `swept` установится в `true`). В результате следующая последовательность приведет к максимальному размеру инкремента

1. `restore (gbak)`
2. `nbackup -b 0`
3. `sweep`
4. `nbackup -b 1`

В пункте 4 инкремент будет содержать все страницы данных, у которых флаг `swept` был установлен в `true`, то есть, этот инкремент будет максимального размера. Поэтому, если это возможно, после `restore` нужно сразу запустить sweeper, а `nbackup` нулевого уровня делать уже после `sweep` (то есть, поменять местами пункты 2 и 3).

Далее, если sweeper будет запускаться между инкрементами, это не приведет ни к каким «лишним» страницам в инкременте. Потому что если на странице появились версии, то она изменена, а если убран мусор, то она тоже изменена, и в следующий инкремент такая страница всё равно должна будет попасть.

## Изменения и дополнения в Firebird 4.0

Появилась возможность использовать Database backup GUID (`gstat -h`) в качестве идентификатора для инкрементного бэкапа.

При этом нужно помнить, что при обычных `-b 0`, `-b 1`, `-b 2` и так далее Database backup GUID каждый раз меняется – именно таким образом Firebird может отслеживать страницы, измененные с момента последнего `-b`, чтобы из них сформировать очередной инкремент.

Пример:

1. Вначале делается обычный полный бэкап базы данных уровня 0  
`nbackup -b 0 <база данных> <копия уровня 0>`  
в этот момент `<база данных>` получает какой-то Database backup GUID.  
пример:  
`nbackup -b 0 e.fdb e.nb0`
2. Берем GUID из вывода `gstat -h` (строка Database backup GUID)

3. Далее, можно выполнить команду

```
nbackup -b <GUID> <база данных> <файл инкремента>
```

В «файл бэкапа» будут изменения, которые произошли с момента выполнения бэкапа уровня 0. То есть, это будет нечто похожее на инкремент уровня 1 (собственно это он и будет).

Пример:

```
nbackup -b {87173C05-23DE-48F1-AEA9-F981F7EEB0DB} e.fdb e.inc
```

! после этой команды database backup guid изменится. И если мы хотим получить новый инкремент, который содержит новые измененные страницы, то для этого надо опять получить Database backup GUID из вывода `gstat -h`, и указать его для очередной итерации.

! если файл инкремента существует, то будет выдана ошибка

*Error creating backup file: e.inc*

*-Файл существует.*

и Database backup GUID при этом изменится, но не попадёт в [rdb\\$backup\\_history](#)! В результате при попытке команды пункта 4 будет выдана ошибка

*Wrong order of backup files or invalid incremental backup file detected, file: e.inc*

Так что для создания инкремента нужно использовать «предыдущий» GUID, который был получен или сразу после `-b 0`, или сразу после успешного `-b {GUID}` (найти такой guid можно в `rdb$backup_history` – это будет либо последний guid для backup level = null, либо для backup level 0).

4. После чего можно взять «файл инкремента», и применить эти изменения к бэкапу уровня 0 (! опция `-INPLACE` должна быть перед `-R`):

```
nbackup -INPLACE -R <копия уровня 0> <файл инкремента>
```

Пример:

```
nbackup -I -R e.nb0 e.inc
```

! повторно выполнить такую же команду нельзя – инкремент уже влит в копию уровня 0

В результате «копия уровня 0» будет обновлена инкрементом «файл бэкапа». Поскольку «копия уровня 0» находится в режиме «backup lock» работать с ней нельзя, однако, если повторять данную операцию периодически (получать инкремент из основной базы и применять его к копии, пункты 2-3-4), то у вас будет в наличии достаточно актуальная копия базы данных, быстро привести которую в рабочее состояние можно командой `nbackup -F`.

Такая схема позволяет гораздо быстрее получить рабочую базу данных из бэкапов, и не требует хранить множество инкрементов разного уровня. Однако, в этом случае нет «истории инкрементов», и вы не сможете восстановить из такого бэкапа базу на момент какого-то предыдущего, произвольно выбранного инкремента (даже если вы их зачем-то решите хранить).

! если копия БД уровня 0 была изменена, то опция `-INCREMENT` повредит эту БД.

! Windows PowerShell может «съесть» фигурные скобки вокруг GUID.

## Изменения и дополнения в Firebird 5.0

Добавились 2 параметра для очистки [лога бэкапов](#) в базе данных:

`-CLEAN_HIST[ORY]`

-KEEP N R[OWS] | D[AYS]

(нигде в документации эти опции не упомянуты, есть только в выводе nbackup и в <https://github.com/FirebirdSQL/firebird/issues/7186> )

Указываются они только с опцией -В (т.е. в процессе бэкапа), и всегда вместе. N для дней или записей должно быть 1 или больше (например, 0 выдаст ошибку *ERROR: Wrong parameter value for switch -keep*)

Пример:

```
nbackup -b 0 e.fdb e.nb0 -clean_hist -k 5 d
```

в истории бэкапов удалятся все записи кроме тех, которые были сделаны в пределах 5 дней от сегодня.

```
nbackup -b 0 e.fdb e.nb0 -clean_hist -k 2 r
```

в истории бэкапов удалятся все записи, кроме двух последних.

Интересно, что если выполнить

```
nbackup -b 1 e.fdb e.nb1 -clean_hist -keep 1 r
```

то пропадет запись истории о бэкапе уровня -0, и при повторном выполнении этой же команды (нбэкап уровня 1 с удалением из истории всех кроме последней записи) будет выдана ошибка *Cannot find record for database "e.fdb" backup level 0 in the backup history*

Не удивительно, т.к. эта запись и правда уже удалена, и предыдущей командой оставлена запись только о бэкапе уровня 1. А такая запись нужна для выстраивания «иерархии» уровней бэкапов. В результате создание инкрементов можно будет возобновить только после очередного полного nbackup -b 0.

**Примечание kdv:** получается, чистить историю бэкапов (и по дням, и по количеству записей) можно только если мы делаем бэкапы уровня 0, но не 1 и далее (иначе, если мы делаем инкременты 1, 2 и т.д. мы можем неверно посчитать количество дней с момента последнего уровня 0).

Или, эти опции можно указывать только при -В 0, но ни в коем случае при -b 1, -b 2..., причем всегда указывая -k 1, т.е. оставляя только текущую запись, и удаляя все предыдущие (потому что см. выше – по дням или количеству мы не угадаем).

## Appendix A: История документа

Историю документа с 21 октября 2005 по 18 сентября 2014 см. в оригинальном документе [https://firebirdsql.org/file/documentation/reference\\_manuals/user\\_manuals/Firebird-nbackup.pdf](https://firebirdsql.org/file/documentation/reference_manuals/user_manuals/Firebird-nbackup.pdf) страницы 18 и 19.

Исправления и дополнения kdv, август 2019, октябрь 2024:

- переписан раздел [Техническая информация](#), исправлены ошибки (например, на стр 17 “... of delta file into main backup”, должно быть of “delta file into main database”.
- Добавление примечания «Важно» в разделе [Инкрементные бэкапы](#).
- Примечания и дополнения в разделе [Кэширование ввода-вывода](#).
- Примечание в разделе [Практическое применение](#).
- Примечание в разделе [Восстановление после -L](#).

- Примечание в разделе [Недостатки nbackup](#).
- Примечания по остальному тексту
- [Изменения и дополнения в Firebird 3.0](#)
- [Изменения и дополнения в Firebird 4.0](#)
- [Изменения и дополнения в Firebird 5.0](#)

## Appendix B: Замечания по лицензии

Содержимое этого документа попадает под действие Public Documentation License Version 1.0 (далее «Лицензия»). Вы можете использовать этот документ только полностью в соответствии с текстом Лицензии. Текст лицензии доступен по ссылкам

<http://www.firebirdsql.org/pdfmanual/pdl.pdf>

и

<http://www.firebirdsql.org/manual/pdl.html>

.

Оригинальный документ называется «СУБД Firebird, утилита nbackup».

Исходный автор документа: Paul Vinkenoog.

Участники: Mark Roteveel, Alexey Kovyazin

Перевод, исправления и дополнения: Кузьменко Дмитрий (iBase.ru)

Авторские права © 2005-2024.