

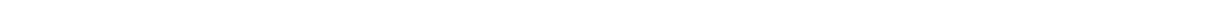


# Утилита nbackup СУБД Firebird

Paul Vinkenoog

8 декабря 2005 - Версия документа 0.1.ru

Перевод на русский язык: Павел Меньщиков



---

## Содержание

Введение .....	3
Функции nbackup: обзор .....	3
Преимущества nbackup .....	3
Ограничения nbackup .....	4
Создание резервных копий и восстановление из них .....	4
Резервная копия всей базы данных .....	4
Инкрементные резервные копии .....	6
Практическое применение .....	8
Читать ли дальше? .....	9
Блокирование и разблокирование .....	9
Блокировка базы данных и самостоятельное резервное копирование .....	10
Восстановление из резервной копии, сделанной после выполнения "nbackup -L" ..	10
Механизмы работы .....	11
А. История документа .....	13
В. Лицензия (англ.) .....	14

---

## Введение

*nbackup* является новой утилитой резервного копирования для СУБД Firebird 2.0. Эта утилита предоставляет возможности, отсутствовавшие в утилите *gbak*, которая была унаследована от СУБД Interbase. Однако она не заменяет полностью возможности последней. Обе программы имеют свои сильные и слабые стороны; вероятнее всего, они будут сосуществовать некоторое время.

## Функции *nbackup*: обзор

С помощью *nbackup* Вы сможете выполнять две различные группы задач:

1. Создание резервных копий (всей базы данных и *инкрементных*) и восстановление из резервных копий. *Инкрементная резервная копия* содержит только изменения со времени создания определенной, ранее созданной резервной копии.
2. Блокирование основного файла базы данных. Таким образом, Вы после этого сможете сами создавать обычные копии или резервные копии с помощью утилит по Вашему выбору. В этом режиме *nbackup* ничего не резервирует, а лишь создает подходящие условия, чтобы Вы могли без каких бы то ни было проблем создавать резервные копии. Также эти условия могут использоваться для восстановления базы данных из резервной копии.

Оба режима могут работать с активной базой данных, не мешая подключенным к базе данным пользователям. Созданная резервная копия базы данных всегда будет отображать состояние базы данных *на момент начала создания резервной копии*. В этой части *nbackup* не отличается от *gbak*.

## Преимущества *nbackup*

- *Оба режима*: высокая скорость (настолько, насколько позволяет оборудование и операционная система), так как *nbackup* не обращается к, собственно, данным. В режиме создания резервной копии содержимое более-менее без всяких изменений записывается в резервный файл.
- *Режим резервирования/восстановления*: сохранение времени и объема свободного места на диске, так как Вам нет необходимости делать резервные копии всей базы данных каждый раз. Это преимущество особенно актуально для баз данных с гигабайтными размерами.
- *Режим блокировки/разблокировки*: свобода в Вашем выборе средств для создания резервных копий, обычных копий базы данных, а также средств компрессии.

## Ограничения nbackup

- nbackup не занимается чисткой (sweep) и сжатием базы данных, как может делать gbak.
- Вы не сможете изменить владельца базы данных при восстановлении базы данных из резервной копии, как могли сделать при помощи gbak.
- nbackup не может создавать резервные копии в *переносимом (transportable) формате*, то есть копии, которые затем можно восстановить на другой платформе или под другой версией СУБД.
- На настоящий момент nbackup не следует использовать на многофайловых базах данных.
- nbackup может работать только с локальными базами данных.

Следующие разделы содержат подробное описание функций nbackup.

## Создание резервных копий и восстановление из них

Для начала, nbackup.exe находится в подпапке bin папки, куда установлена СУБД Firebird. Например, типичным расположением является C:\Program Files\Firebird\Firebird\_2\_0\bin (Windows) или /opt/firebird/bin (Linux). Как и у большинства утилит, распространяемых с СУБД Firebird, у nbackup нет графического интерфейса; Вы запускаете программу из командной строки (или из командного файла, или из другой программы).

### Резервная копия всей базы данных

#### Содание резервной копии всей базы данных

Для создания резервной копии всей базы данных используйте следующий синтаксис командной строки (перенос на следующую строку сделан исключительно из эстетических соображений):

```
nbackup [-U <пользователь> -P <пароль>]
        -B 0 <база_данных> [<резервный_файл>]
```

Например:

```
C:\Databases> nbackup -B 0 inventory.fdb inventory_1-Mar-2006.nbk
```

Примечания:

- Параметр `-B` означает создание резервной копии. *Уровень резервной копии 0* означает создание резервной копии всей базы данных. Уровни резервных копий больше 0 используются для создания инкрементных резервных копий; это будет рассмотрено далее.
- Вместо имени файла базы данных Вы можете указать псевдоним (`alias`, из файла `aliases.conf`).
- Вместо имени файла резервной копии Вы также можете указать `stdout`. Это перенаправит резервную копию в стандартный поток вывода, откуда Вы сможете перенаправить ее, например, на ленточный накопитель или на вход утилиты для сжатия получаемой резервной копии.
- Параметры `-U` (`user`, имя пользователя) и `-P` (`password`, пароль) могут быть опущены (не указываться):
  - если Вы зарегистрированы в системе как администратор (`root`, `Administrator...`), или
  - если установлены переменные окружения `ISC_USER` и `ISC_PASSWORD`.

Для понятности и краткости эти параметры не указаны в приводимых примерах.

- Все параметры (`-B`, `-U` и `-P`) можно указывать в произвольном порядке. Естественно, за каждым параметром должен(ны) следовать его аргумент(ы). В случае с параметром `-B` есть три аргумента: уровень резервной копии, база данных и файл резервной копии - в этом порядке!
- Если параметр `-B` указан последним, Вы *можете* не указывать имя файла резервной копии. В этом случае, `pbackup` построит имя файла на основе имени базы данных, уровне резервной копии и текущем дате и времени. Это может привести к коллизии имен файлов (и неудавшемуся резервному копированию), если две команды резервирования одного уровня вызываются в одну и ту же минуту.

#### **Внимание**

*Не* используйте `pbackup` для многофайловых баз данных. Это может привести к повреждениям базы данных и потере данных - `pbackup` не будет возражать против выполнения действий над многофайловой базой данных.

### **Несколько слов о внутренних механизмах**

На заметку: то, что здесь будет описано, не является необходимыми знаниями для использования `pbackup`. Это описание дает грубое представление о том, что происходит при работе программы `pbackup` с параметром `-B`:

1. Прежде всего, основной файл базы данных блокируется установкой внутреннего флага состояния. С этого момента абсолютно все изменения в базе данных записываются во временный файл, называемый файлом разницы (`difference file`) или *файлом дельты*.

2. После этого создается резервная копия. Это не обычная копия файла базы данных - восстановление из полученной копии необходимо производить также при помощи nbackup.
3. По завершении резервирования содержимое файла дельты объединяется с основным файлом базы данных. После этого база данных разблокируется (флаг возвращается в «нормальное» состояние) и файл дельты удаляется.

Функциональность шагов 1 и 3 достигается введением двух новых операторов SQL: ALTER DATABASE BEGIN BACKUP и ALTER DATABASE END BACKUP. Вразрез с указанным в операторах, они *не* ведут к созданию резервной копии, они лишь создают условия, с которыми можно безопасно создать резервную копию основного файла базы данных. Чтобы прояснить: Вам не нужно употреблять указанные операторы самостоятельно и явно; nbackup сделает это за Вас в нужное время.

## Восстановление из резервной копии всего файла базы данных

Резервная копия всей базы данных восстанавливается следующим образом (перенос на следующую строку сделан исключительно из эстетических соображений):

```
nbackup [-U <пользователь> -P <пароль>]
        -R <база_данных> [<резервный_файл>]
```

Например:

```
C:\Databases> nbackup -R inventory.fdb inventory_1-Mar-2006.nbk
```

Примечания:

- Вам не нужно указывать уровень при восстановлении.
- При восстановлении параметр *-R* должен быть указан последним по причинам, которые будут описаны позже.
- Если указанная база данных уже существует и нет активных соединений, она будет перезаписана без предупреждения! Если есть активные соединения, восстановление не состоится и Вы получите сообщение об ошибке.
- Здесь также Вы можете не указывать имя файла резервной копии. Если Вы его опустите, nbackup спросит Вас об этом позже. *Однако на текущий момент разработки СУБД Firebird 2 (стадия alpha 3) это приведет к ошибке (по крайней мере под Windows) и неудавшемуся восстановлению.*

## Инкрементные резервные копии

### Создание инкрементных резервных копий

---

Для создания инкрементной («дифференциальной») резервной копии необходимо указать уровень резервной копии больше 0. Инкрементная резервная копия уровня  $N$  содержит изменения базы данных с момента создания последней резервной копии уровня  $N-1$ .

Примеры:

Через день после создания резервной копии всей базы данных (уровня 0) Вы создаете резервную копию уровня 1:

```
C:\Databases> nbackup -B 1 inventory.fdb inventory_2-Mar-2006.nbk
```

Эта резервная копия будет содержать только изменения базы данных за последний день.

Через день Вы вновь решили сделать резервную копию уровня 1:

```
C:\Databases> nbackup -B 1 inventory.fdb inventory_3-Mar-2006.nbk
```

Эта копия будет содержать изменения за последние *два* дня, то есть с момента создания резервной копии всей базы данных, а не только с момента создания предыдущей инкрементной копии уровня 1.

Через пару часов Вы создаете резервную копию уровня 2:

```
C:\Databases> nbackup -B 2 inventory.fdb inventory_3-Mar-2006_2.nbk
```

Эта резервная копия будет содержать изменения только с момента создания последней резервной копии уровня 1, то есть только за последние несколько часов.

#### **Замечание**

Все примечания, сделанные по поводу создания резервной копии всей базы данных, применимы и к созданию инкрементных резервных копий.

#### **Внимание**

Еще раз: не используйте nbackup для многофайловых баз данных.

## **Восстановление из инкрементных резервных копий**

При восстановлении базы данных из инкрементных резервных копий Вы должны обеспечить наличие полной цепочки инкрементных резервных копий, начиная с уровня 0 и до уровня, которым Вы хотите завершить. База данных всегда строиться с самой первой резервной копии уровня 0, шаг за шагом.

Формальный синтаксис:

```
nbackup [-U <пользователь> -P <пароль>]  
-R <база_данных> [<резервная_копия0>  
[<резервная_копия1> [...] ] ]
```

Таким образом, восстановление для предыдущего примера до уровня 2 будет выглядеть так:

```
C:\Databases> nbackup -R inventory.fdb inventory_1-Mar-2006.nbk
                inventory_3-Mar-2006.nbk inventory_3-Mar-2006_2.nbk
```

Перенос на новую строку сделан здесь исключительно из эстетических соображений - Вам необходимо вводить команду в командной строке полностью, и нажать **Enter** только в конце.

Примечания (дополнительно к [примечаниям по восстановлению из резервной копии всей базы данных](#)):

- Так как программа не может знать заранее количество указанных после параметра `-R` имен файлов (уровень при восстановлении не указывается), nbackup считает все аргументы после параметра `-R` именами файлов с резервными копиями. По этой причине никакие другие параметры (`-U` или `-P`) не могут следовать за списком файлов параметра `-R`.
- Не существует формального ограничения на уровень резервной копии, однако на практике редко имеет смысл создавать копии уровней больше 3 или 4.

### **Несвязанные ссылки**

Что произойдет, если Вы нечаянно пропустите файл с инкрементной резервной копией в цепочке восстановления, или укажете набор файлов, которые не являются одной цепочкой? Представьте, что Вы по ошибке указали `inventory_2-Mar-2006.nbk` вместо `inventory_3-Mar-2006.nbk` в вышеприведенном примере. Обе резервные копии являются копиями уровня 1, поэтому в обоих случаях у Вас получится замечательная последовательность уровней «0, 1, 2». Но файл уровня 2 является инкрементной резервной копией для инкрементной резервной копии уровня 1 от 3 марта, а не от 2 марта.

К счастью, такие ошибки никогда не приведут к неверно восстановленной базе данных. Каждый файл с резервной копией имеет уникальный идентификатор. Более того, каждый резервный файл уровня 1 и выше содержит идентификатор того файла, на котором он основан. При восстановлении nbackup проверяет эти идентификаторы; если где-то в указанной цепочке обнаруживается неверная ссылка, операция восстановления не производится и Вы получите сообщение об ошибке.

## **Практическое применение**

Основанная на nbackup инкрементная схема резервирования может выглядеть следующим образом:

- Каждый месяц создается резервная копия всей базы данных (уровня 0);
- Каждую неделю делается инкрементная резервная копия уровня 1;
- Каждые сутки создается инкрементная резервная копия уровня 2;
- Каждый час создается инкрементная резервная копия уровня 3.

Поскольку все резервные копии сохраняются, Вы сможете восстановить базу данных в любое

состояние с точностью до часа. При каждом восстановлении используется максимум до четырех резервных файлов. Разумеется, Вам необходимо так планировать процесс создания резервных копий, что наибольшие из них (требующие больше времени) создаются во время наименьшей нагрузки на СУБД со стороны пользователей. В указанной схеме уровни 0 и 1 могут создаваться по выходным, а уровень 2 - в ночное время.

Если Вы не хотите хранить все созданные резервные копии, Вы можете спланировать схему удаления ненужных копий:

- Резервные копии уровня 3 удаляются после 8 дней хранения с момента создания;
- Резервные копии уровня 2 - после месяца;
- Резервные копии уровня 1 - после полугодия;
- Резервные копии уровня 0 (всей базы данных) - после двух лет, но первую резервную копию всей базы данных каждого года нужно сохранить.

Конечно, приведенные схемы являются лишь примером. Что будет подходящим в конкретном случае, зависит от приложения, размера базы данных, активности пользователей и т.д.

## **Читать ли дальше?**

Сейчас Вы знаете все, что нужно, для того, чтобы создавать резервные копии базы данных и производить восстановление базы данных из резервных копий с помощью nbackup. Если Вы хотите использовать другие утилиты для создания резервных копий баз данных Firebird, то читайте следующие разделы.

Если у Вас нет желания вникать в тонкости, удачи Вам в обычной работе с nbackup!

## **Блокирование и разблокирование**

Если Вы предпочитаете использовать какие-то другие утилиты для создания резервных копий базы данных или просто делать обычную копию базы данных как резервную, то в игру вступает режим блокировки/разблокировки программы nbackup. «Блокировка» в данном случае означает, что основной файл базы данных временно замораживается, а не невозможность внесения изменений в базу данных. Как и в режиме резервирования, изменения фиксируются во временном файле дельты; при разблокировании файл дельты объединяется с основным файлом базы данных.

В качестве напоминания: nbackup.exe находится в подпапке bin папки, куда установлена СУБД Firebird. Типичными местонахождениями, например, являются C:\Program Files\Firebird\Firebird\_2\_0\bin (Windows) или /opt/firebird/bin (Linux). У утилиты нет графического интерфейса; Вы запускаете ее из командной строки (или из пакетного файла, или из другого приложения).

## **Блокировка базы данных и самостоятельное резервное копирование**

Типичным сценарием является следующий:

1. Блокировать базу данных с помощью параметра `-L` (Lock):

```
nbackup [-U <пользователь> -P <пароль>] -L <база_данных>
```

2. Теперь можно создать резервную копию, сжать файл базы данных (и много еще чего можно делать с содержимым файла базы данных), используя Ваши любимые программы. Простое копирование файла также допустимо.

3. Разблокировать базу данных с помощью параметра `-N` (uNlock):

```
nbackup [-U <пользователь> -P <пароль>] -N <база_данных>
```

Последняя команда также приведет к объединению файла дельты, куда записывались все изменения (мета)данных за время блокировки, с основным файлом базы данных.

Созданная резервная копия будет содержать данные, которые являлись актуальными на момент начала блокировки; эти данные не зависят от длительности блокировки, а также от периода времени, прошедшего с начала блокировки до момента начала резервного копирования.

### **Внимание**

Что применимо к резервированию/восстановлению, также применимо и к блокированию/разблокированию: не используйте блокирование/разблокирование на многофайловых базах данных. Пока ситуация не изменится, не подпускайте nbackup к многофайловым базам данных!

## **Восстановление из резервной копии, сделанной после выполнения "nbackup -L"**

Копия заблокированной базы данных является так же заблокированной, поэтому Вы не сможете просто использовать копию как рабочую базу данных. В случае, если Ваша исходная база данных повреждена или утеряна, и нужно восстановить базу данных из копии, сделанной Вами самостоятельно, действуйте следующим образом:

1. Разархивируйте/скопируйте/восстановите файл базы данных с помощью используемых Вами утилит.
2. Теперь разблокируйте базу данных, но *не* с параметром `-N`, а с параметром `-F` (Fixup):

```
nbackup -F <база_данных>
```

Почему существует два параметра командной строки,  $-N$  и  $-F$ ?

- При использовании параметра  $-N$  сначала определяется наличие любых изменений с момента блокирования базы данных (после использования параметра  $-L$ ) и производится объединение временного файла дельты и основного файла базы данных. После этого база данных переводится в нормальный режим чтения/записи, а временный файл удаляется.
- При использовании параметра  $-F$  только изменяется в «нормальное» значение флага состояния самостоятельно восстановленной базы данных.

Итак, Вы используете:

- параметр  $-N$  после *создания* резервной копии своими силами (для возвращения флага состояния после ранее выполненного блокирования файла с параметром  $-L$ );
- параметр  $-F$  после самостоятельного *восстановления* из такой резервной копии.

#### Замечание

Не очень хорошо получилось, что последний параметр  $-F$  назван по слову Fixup (поправить): его предназначение не исправлять что-либо, а только *разблокировать* базу данных. Параметр  $-N$  (uNlock, разблокировать), с другой стороны, не только разблокирует базу данных, но и вносит в нее некоторые правки (внедряет сделанные изменения в базу данных). Однако, нам придется работать с тем, что есть.

## Механизмы работы

На заметку: то, что здесь будет описано, не является необходимыми знаниями для использования nbackup, но предоставляет Вам дополнительную информацию, которая поможет Вам глубже понять использование различных параметров командной строки.

**nbackup -L** делает следующее:

1. соединяется с базой данных;
2. запускает транзакцию;
3. выполняет оператор ALTER DATABASE BEGIN BACKUP (этот оператор обсуждался в разделе [дополнительной информации по nbackup -B](#));
4. завершает (commit) транзакцию;
5. отсоединяется от базы данных.

**nbackup -N** делает те же шаги, но выполняет оператор ...END BACKUP на шаге 3.

**nbackup -F** работает следующим образом:

1. открывается восстановленный файл базы данных;
2. в файле базы данных флаг состояния изменяется с заблокированного (nbak\_state\_stalled) в нормальное (nbak\_state\_normal) состояние;
3. файл закрывается.

**Замечание**

pbackup *-F* работает на файловом уровне, а поэтому может быть выполнена даже при отсутствии работающего сервера Firebird. Любой из дополнительно указанных параметров *-U* и *-P* полностью игнорируются.

## История документа

Точная история изменения исходного файла фиксируется (или будет) в модуле `manual` нашего дерева CVS; обратитесь к [http://sourceforge.net/cvs/?group\\_id=9028](http://sourceforge.net/cvs/?group_id=9028)

### История переиздания

0.1	21 окт 2005	PV	Первая редакция.
0.1.ru	8 дек 2005	PM	Документ переведен на русский язык.

## Лицензия (англ.)

The contents of this Documentation are subject to the Public Documentation License Version 1.0 (the «License»); you may only use this Documentation if you comply with the terms of this License. Copies of the License are available at <http://www.firebirdsql.org/pdfmanual/pdl.pdf> (PDF) and <http://www.firebirdsql.org/manual/pdl.html> (HTML).

The Original Documentation is titled *Firebird's nbackup tool*.

The Initial Writer of the Original Documentation is: Paul Vinkenoog.

Copyright (C) 2005. All Rights Reserved. Initial Writer contact: paulvink at users dot sourceforge dot net.

Contributor: Pavel Menshchikov - Russian translation.

Copyright (C) 2005. All Rights Reserved. The translator contact: developer at ls-software dot ru.