

# 5 First Steps To Better Firebird Performance

A.Kovyazin, [ak@ib-aid.com](mailto:ak@ib-aid.com), E.Gregorio, [edson.gregorio@mqfs.com.br](mailto:edson.gregorio@mqfs.com.br)

Version 1.0, by 2024-05-15

# Table of Contents

1. Step 1 of 5: Check OS performance settings on your Firebird server .....	2
1.1. For Windows .....	2
1.1.1. Optional steps for Windows .....	4
1.2. For Linux .....	4
1.2.1. Optional steps for Linux .....	5
2. Step 2 of 5: Create the optimal firebird.conf for your Firebird server .....	6
2.1. What is the problem with the default configuration? .....	6
2.2. Create optimal Firebird configuration .....	6
2.3. Very important thing! .....	7
3. Step 3 of 5: Check Disk Performance on Firebird Server .....	9
3.1. CrystalDiskMark (Windows) .....	9
3.2. The Simple Insert Update Delete Test .....	10
3.3. How to run Simple Insert Update Delete Test .....	10
3.3.1. Speed test with HQbird .....	10
3.3.2. Speed test on server with any Firebird.. .....	11
3.3.3. How to interprete results .....	12
3.4. The possible reasons why disk can be slow .....	13
4. Step 4 of 5: Check your Firebird database indices for common problems .....	14
4.1. Check non-active or non-activated indices .....	14
4.2. Recalculate selectivity (statistics) of indices in your Firebird database .....	14
4.3. How to check indices state and recalculate statistics with HQbird .....	16
5. Step 5 of 5. Choose proper page size for your Firebird database .....	17
5.1. A bit of theory .....	17
5.2. Default page size .....	18
5.3. Database size .....	18
5.4. Depth of indices .....	18
5.5. Size of compressed records .....	19
5.6. Recalculate configuration after change of page size .....	20
6. Summary .....	21

[\(A versão em português está aqui\)](#)

Below you will find 5 essential steps to make your Firebird database faster and better. In these steps, we'll give you clear instructions on what to look for and how to fix or improve it. You should start Firebird performance optimizations with these 5 steps.

# Chapter 1. Step 1 of 5: Check OS performance settings on your Firebird server

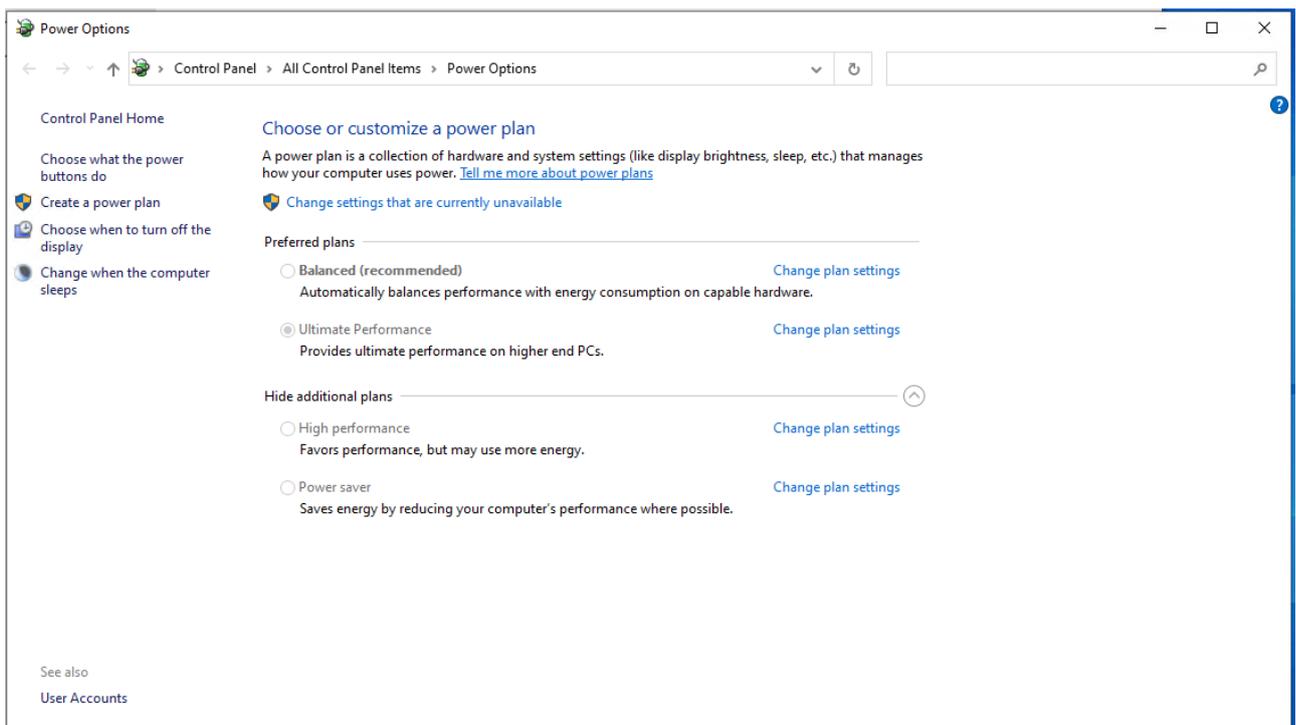
To run FirebirdSQL in the most efficient way, make sure you have set the necessary settings of the Operating System.

## 1.1. For Windows

In general, Windows users need to do the following 3 steps:

1. Enable “Ultimate Performance” or “High Performance”

Change your power plan to “High Performance” (or Ultimate, if you have it). This can boost your CPU performance by up to 20%, because it makes your CPU cores run faster with higher energy consumption. To change your power plan, go to Windows energy settings (“Power Options”) and choose “High performance” (instead of “Balanced”, which is the usual one).



No need to restart Firebird or reboot server, the effect is applied immediately.

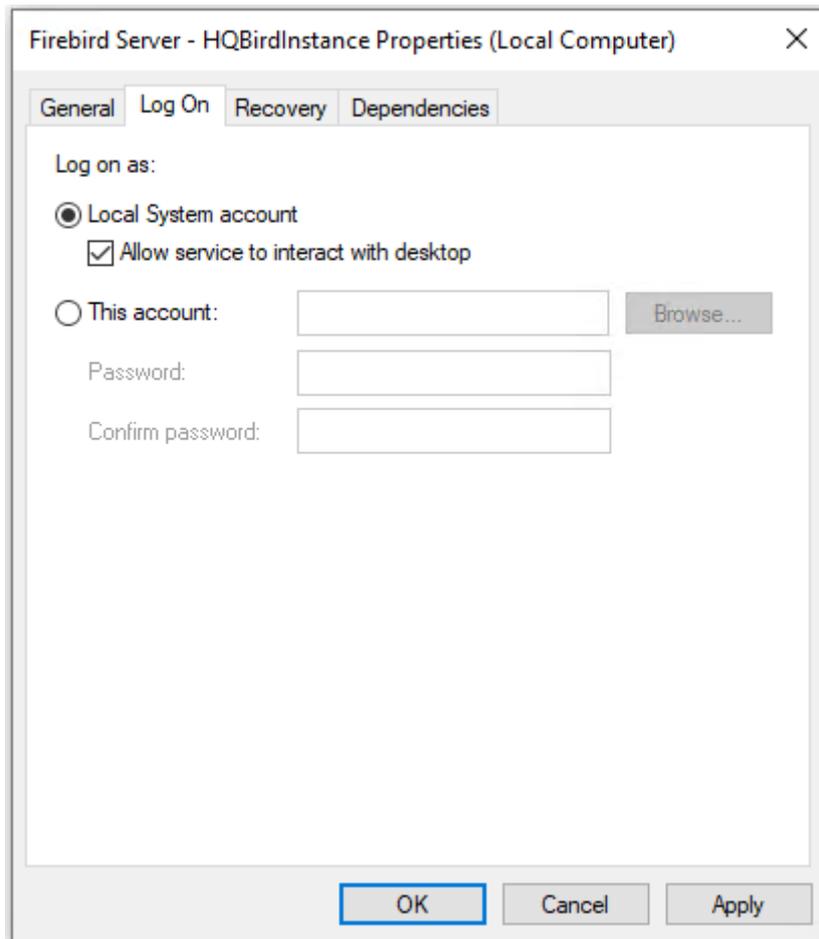


Please note, that Windows Update sometimes resets this setting to the default “Balanced”.

2. Enable “Allow service to interact with desktop” option for Firebird service

This option is absolutely necessary for Firebird 2.5 with Classic architecture, and will be helpful for other architectures too, especially for the Firebird databases with high number of connections.

To see the effect, it is necessary to restart all instances of Firebird.



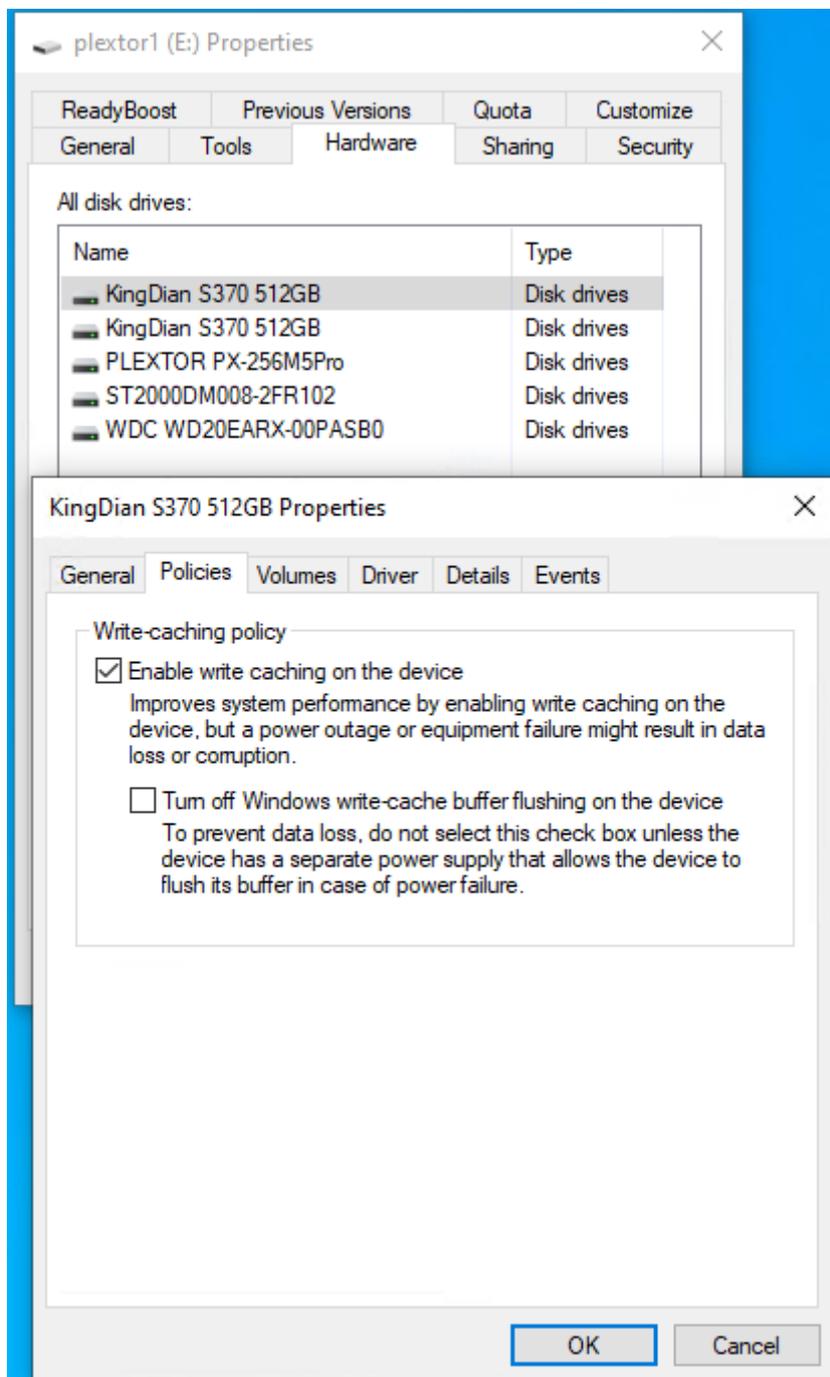
### 3. For all disks, check Write-Caching Policy

If you don't have RAID or SAN or other complex storage device, make sure that Windows has enabled write caching. Right-click on disk, select Properties, and tab Policies. The option "Enable write caching on the device" should be enabled.

No need to reboot Windows, until it will be explicitly required.



If you do have RAID— need to configure it correctly. We will devote another article to explain how to configure popular RAIDs.



### 1.1.1. Optional steps for Windows

The following recommendations are based on practical findings of IBSurgeon support:

1. Swap file size. The recommended option is to let Windows define the swap size. If you decide to limit the swap file, make sure it is set to at least 32Gb. Never turn off swap!
2. Uptime. Reboot your Windows server or VM at least once per 180 days.

## 1.2. For Linux

On Linux, there are 2 mandatory steps for Firebird servers:

1. Increase `MaxOpenFiles`

Firebird instance with high number of connections will require to increase number of open files (handles).

Check the value of `MaxOpenFile` it with the following command:

```
cat /proc/firebird_process_id/limits
```

The default is 4096, increase it if necessary to 49999.

Set it in Firebird service file `LimitNOFILE=49999`

Usually, the Firebird service files is located in `/usr/lib/systemd/system/firebird-superserver.service`, or in `/usr/lib/systemd/system/firebird`

Firebird shipped with HQbird (in versions 2020+) already has this tuning.

## 2. Check and increase VMA, if necessary

Firebird instances with high number of connections, with architectures SuperServer and SuperClassic can require many Virtual Memory Areas (VMA).

Check VMA count with the following command

```
cat /proc/sys/vm/max_map_count
```

The default is 64K, it should be set to 250K.

To increase it, add the following line to `/etc/sysctl.conf` the line `vm.max_map_count=262144`

To apply, reboot server, or, to apply immediately, do

```
sysctl -w vm.max_map_count=262144
```



Please note, to make this setting permanent, need change of `sysctl.conf`!

### 1.2.1. Optional steps for Linux

1. Check that you have enough space in `/tmp`. We recommend to have at least 50Gb. Firebird stores temporary files in `/tmp` and `/tmp/firebird`, and they can be rather large. If you use Firebird HQbird with `BlobTempSpace` option, there will be stored Blob temp files.
2. Uptime. Reboot your server at least once per 90 days.
3. Swap. Never disable swap on Linux.

# Chapter 2. Step 2 of 5: Create the optimal `firebird.conf` for your Firebird server

## 2.1. What is the problem with the default configuration?

Are you using the default `firebird.conf` file that comes with the regular (vanilla) Firebird, installed from the standard installer or Linux repo? If so, we have some bad news for you. Your `firebird.conf` file is not the best one for your Firebird database.

The default `firebird.conf` file is made to work on the smallest and simplest hardware or VM. But that's not what most production servers need.

If you are using HQbird with Firebird, you are doing better. HQbird gives you a `firebird.conf` file that is better for the average server.

## 2.2. Create optimal Firebird configuration

Do you want to make your `firebird.conf` (and `databases.conf`) files better for your hardware or VM? That's a smart idea. We have a free service that can help you with that. It's called "Configuration Calculator for Firebird", and it can make your Firebird database use your resources better. The service will make the better than standard configuration for you, and, more important, help you avoid common mistakes in Firebird configuration.

If you want to make your configuration even better, we can do more things for you in the frames of our [Optimization Services](#), like analyzing your queries, your database structure, and more.

To try the Configuration Calculator, just click this link: <https://cc.ib-aid.com/democalc.html> or, better, register in <https://cc.ib-aid.com>, to access other tools for Firebird.

Select Firebird version, architecture, and then details of your hardware/VM (RAM/CPU cores), and characteristics of database and load: database size, database page size, and click "Create":

The screenshot shows a web browser window with the URL `https://cc.ib-aid.com/democalc.html?lang=en_EN`. The page title is "Calculate configuration". On the left sidebar, there are icons for "Home", "Traces files", and "Get config". The main content area is divided into two columns. The left column contains form fields for:
 

- Choose your Firebird version\*:  Firebird 4 (vanilla),  Firebird 4 HQbird,  Firebird 3 (vanilla),  Firebird 3 HQbird,  Firebird 2.5 (vanilla),  Firebird 2.5 HQbird
- Choose your architecture\*:  SuperServer,  Classic,  SuperClassic
- RAM(Gb):
- Total number of users:
- Number of CPU cores:
- Database page size:
- Database size(Gb):
- Main database name:
- Main database path:

 A green "Create" button is at the bottom left. The right column shows two code blocks:
 

- firebird.conf**: Contains configuration for the Firebird SuperServer, including cache settings, lock table options, and authentication plugins. A "Copy" button is next to it.
- databases.conf**: Shows the configuration for the security database, including the path to the security3.fdb file and specific settings for the 'testdb' database. A "Copy" button is next to it.

Next, you need to copy and paste the results into your `firebird.conf` and `databases.conf` files.

But before you do that, make sure you delete everything that's already in those files, even the comments. This way, you won't have any problems with the old and new settings.

## 2.3. Very important thing!

Make sure you don't have value for **Page Buffers** line in the header page of your database!

If you have to ensure it, run in terminal (command prompt):

```
/opt/firebird/bin/gstat -h /path/to/database.fdb
```

or (on Windows)

```
<your Firebird bin folder>\gstat.exe -h Disk:Pathtodatabase.fdb
```

The output will look like this:

```
Gstat execution time Mon Nov 28 11:25:03 2022

Database header page information:
Flags 0
Generation 5914
System Change Number 0
Page size 16384
ODS version 12.0
Oldest transaction 6194
Oldest active 6195
Oldest snapshot 6195
Next transaction 6220
Sequence number 0
Next attachment ID 63
Implementation HW=AMD/Intel/x64 little-endian OS=Windows CC=MSVC
Shadow count 0
Page buffers 0
Next header page 0
Database dialect 3
Creation date Nov 24, 2022 16:42:58
Attributes
Variable header data:
Sweep interval: 20000
Database GUID: {260809FC-FFF7-488F-7ABF-FEF0A84F6900}
*END*
Gstat completion time Mon Nov 28 11:25:03 2022
```

Check that line with parameter **Page Buffers** is 0.

If it is set to other value, it will override the settings in [firebird.conf](#) and [databases.conf](#).

To use values from configuration, set **Page Buffers** in header to 0 with command

```
gfix -buff 0 /path/to/database.fdb -user SYSDBA -pass masterkey
```

To apply new configuration, restart all instances of Firebird.

As a result, you will see improved performance of your Firebird instance.

# Chapter 3. Step 3 of 5: Check Disk Performance on Firebird Server

One of the common causes of Firebird database slowness is when your disk is not fast enough: your disk is where your Firebird database stores and reads your data, so it needs to be quick. Let's test how well your disk works with your Firebird database.

## 3.1. CrystalDiskMark (Windows)

There are several universal tools to measure the disk performance in the common way, they give you the estimation in MB/sec or in IOPS. The most popular is CrystalDiskMark (<https://crystalmark.info/>).

In our experience, CrystalDiskMark gives adequate estimation of disk speed and can be used as a starting point. The example of CrystalDiskMark output in the text format:

```
-----
CrystalDiskMark 8.0.4 x64 (C) 2007-2021 hiyohiyo
                        Crystal Dew World: https://crystalmark.info/
-----
* MB/s = 1,000,000 bytes/s [SATA/600 = 600,000,000 bytes/s]
* KB = 1000 bytes, KiB = 1024 bytes

[Read]
SEQ  1MiB (Q= 8, T= 1): 2334.517 MB/s [ 2226.4 IOPS] < 3588.90 us>
SEQ  1MiB (Q= 1, T= 1): 1799.226 MB/s [ 1715.9 IOPS] < 582.13 us>
RND  4KiB (Q= 32, T= 1): 411.333 MB/s [ 100423.1 IOPS] < 318.20 us>
RND  4KiB (Q= 1, T= 1): 25.074 MB/s [ 6121.6 IOPS] < 163.02 us>

[Write]
SEQ  1MiB (Q= 8, T= 1): 999.476 MB/s [ 953.2 IOPS] < 8341.94 us>
SEQ  1MiB (Q= 1, T= 1): 1543.671 MB/s [ 1472.2 IOPS] < 675.37 us>
RND  4KiB (Q= 32, T= 1): 83.297 MB/s [ 20336.2 IOPS] < 1564.70 us>
RND  4KiB (Q= 1, T= 1): 70.380 MB/s [ 17182.6 IOPS] < 57.63 us>

Profile: Default
  Test: 64 GiB (x5) [C: 89% (828/931GiB)]
  Mode: [Admin]
  Time: Measure 5 sec / Interval 5 sec
  Date: 2022/11/29 17:32:33
  OS: Windows 11 [10.0 Build 22621] (x64)
```

However, it is not enough to have a warranty that disk is really fast for Firebird, because the test is very synthetic.

## 3.2. The Simple Insert Update Delete Test

In order to measure the performance of disk from the Firebird, IBSurgeon has developed the script, which creates the database, inserts 1 million of records, then updates these records, and then deletes them.

The script is executed by `isql.exe` standard tool, with the actual configuration of Firebird.

Essentially, it is single-thread test which measures Firebird performance for the case of a “peak” processing operation — the result of the test is a database file with size 3.6 Gb.

The result is presented in the form or number of milliseconds to perform Insert 1 Mln, Update 1 Mln, Delete 1 Mln. records separately for the operation and for the commit time.

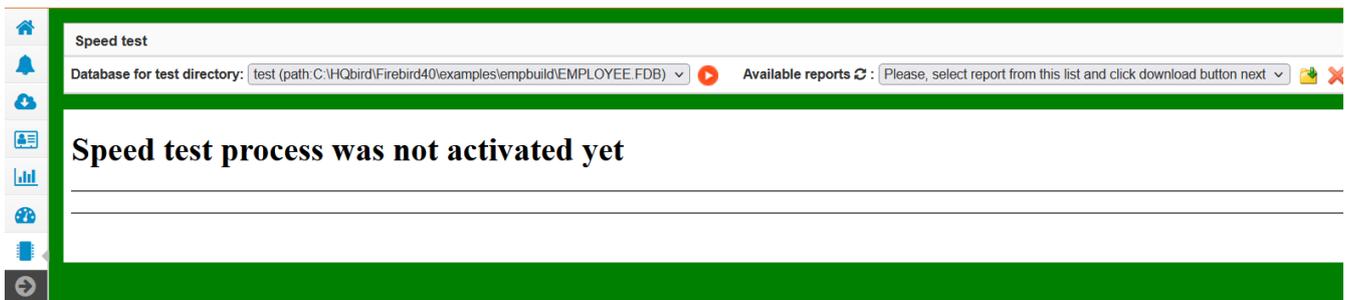
Since the creation of the script, people have run it on the hundreds of servers (many people from the community have run the test on various servers and sent its results), and now there is the table and the diagram with results.

To facilitate comparison, the result is presented in the format of Inserts/seconds, Updates/seconds, Deletes/second.

## 3.3. How to run Simple Insert Update Delete Test

### 3.3.1. Speed test with HQbird

You can use HQbird to test your disk speed with “Simple IUD Test”. Go to its web-console and look for the icon that says “Speed Test” on the right side. Then, choose the folder where you want to do the test from the drop-down menu. And then, just click Start. That’s it.



After 5-10-20 minutes, you will see results of a speed test:

**Speed test on "test" at Wed Feb 07 21:36:12 CET 2024 in the directory of database C:\HQbird\Firebird40\examples\empbuild\EMPLOYEE.FDB**

**Speed test process is complete!**

Parameter	Value				
Source database name:	test				
Source database file:	C:\HQbird\Firebird40\examples\empbuild\EMPLOYEE.FDB				
Dataguard DB ID:	2402071441_EMPLOYEE_FDB				
Server version:	WI-V4.0.4.2977 Firebird 4.0 HQbird				
Speed Test Database:	C:\HQbird\Firebird40\examples\empbuild\hqbird_speedtest.fdb.temp				
Result Size:	3815243776				
Started at	2024-02-07 21:36:12				
Finished at	2024-02-07 21:42:09				
Score value for Inserts:	9711 rec/sec				
Score value for Updates:	6662 rec/sec				
Score value for Deletes:	9805 rec/sec				
Test result raw data	<table border="1"> <thead> <tr> <th>INSERTED_ROWS</th> <th>ELAP_MS</th> </tr> </thead> <tbody> <tr> <td>1000000</td> <td>99871</td> </tr> </tbody> </table>	INSERTED_ROWS	ELAP_MS	1000000	99871
INSERTED_ROWS	ELAP_MS				
1000000	99871				

### 3.3.2. Speed test on server with any Firebird.

In order to run the test on any Firebird server, copy-paste the script from the web-site: [https://ib-aid.com/en/simple-insert-update-delete-test-for-firebird/#how\\_to\\_run\\_firebird\\_performance\\_test](https://ib-aid.com/en/simple-insert-update-delete-test-for-firebird/#how_to_run_firebird_performance_test) and save it to the file, for example, `/opt/script1.sql` or `c:\temp\script1.sql`.

Then, edit the path in the 1<sup>st</sup> line of the script to the test database to be created — it should be on the same drive where main database is located:

```
create database "localhost:<DISK>:<PATH>inserttest4.fdb"
user "SYSDBA" password "masterkey" page_size 16384;
```

or, for Linux

```
create database "localhost:/<path>/inserttest4.fdb"
user "SYSDBA" password "masterkey" page_size 16384;
```

Avoid situation when tested disk is loaded with a long operation like backup or copying of large volumes of data, or simply have a peak number of users. The best option is to run script in the exclusive mode without users.

Run script from the command prompt with command like this:

```
/opt/firebird/bin/isql -i /opt/script1.sql
```

or

```
"C:\Program Files\Firebird\Firebird_4_0\isql.exe" -i .\script1.sql
```

The output will be on the screen (only 1<sup>st</sup> part for INSERTs is shown below):

```
C:\HQbird\Firebird30> isql -i c:\Temp\test1.sql
Use CONNECT or CREATE DATABASE to specify a database
INSERTED_ROWS 1000000
ELAP_MS 58890
Current memory = 702384256
Delta memory = 3215392
Max memory = 704756112
Elapsed time= 58.894 sec
Buffers = 40960
Reads = 9
Writes = 88015
Fetches = 25332353
Current memory = 702342416
Delta memory = -41840
Max memory = 704756112
Elapsed time= 4.846 sec
Buffers = 40960
Reads = 1
Writes = 40254
Fetches = 2
```

### 3.3.3. How to interpret results

Please note, it is necessary to sum operation's time and its commit time, and then divide 1000000 to the total time.

For the example above,

```
INSERT SPEED = 1000000 / (58.984+4.846) = 15666 inserts /seconds
```

Results in the form Operation per second — for example, 6000 Inserts/second, 5200 Updates/second, 7300 deletes/second, can be compared against the table/graph of performance measurements.

Table is here: [https://ib-aid.com/images/testresult/insert\\_update\\_delete\\_results.xls](https://ib-aid.com/images/testresult/insert_update_delete_results.xls)

If you see the results above the middle of the graph/table, it means that disk speed is Ok, and reason for slowness will be in another area than hardware.

Another thing to check is the time to commit. If you see that total test time contains more than 20% of the commit, it means that there could be problem with write cache.

If you see that result of the test is below the average, and you have SSD disks, it means that there is some problem with disk configuration or driver.

### **3.4. The possible reasons why disk can be slow**

1. Wrong file system for drive where Firebird database resides
2. Absence of Battery Backup Unit for RAID (or discharged battery)
3. Absence of suitable driver (or driver is disabled by the vendor to prevent you from using consumer-grade disc on server operating system)
4. Limit of IOPS for VM (for example, Azure has very low IOPS limits for general purpose virtual machines)
5. Active backup process or other processing (copying, rearrangement, etc) of large data on disk.
6. Disk is going to die: bad blocks, overheating, etc. Always check S.M.A.R.T data for the disk

# Chapter 4. Step 4 of 5: Check your Firebird database indices for common problems

In order to keep Firebird database in a good shape, all indices should be healthy and with up-to-date statistics, and effective.

Let's see how to perform these checks.

## 4.1. Check non-active or non-activated indices

Firebird database indices can exist in one of three states: active, inactive, or non-activated.

1. An active index is a regular index that optimizes query performance.
2. An inactive index is an index that has been explicitly disabled by the `ALTER INDEX ... INACTIVE` command. An inactive index will be not used in queries.
3. A non-activated index is an index that was intended to be enabled during the `gbak` restore process, but the process was terminated prematurely (typically due to an error). Non-activated indices should not normally be present in the database. A non-activated index will be not used in queries.

To inspect the state of indices in a Firebird database, execute the following query:

```
SELECT count(*) FROM RDB$INDICES WHERE RDB$INDEX_INACTIVE<>0;
```

If the result is 0, all indices are active. If results is not 0, do the following query to see the list of problematic indices:

```
SELECT RDB$INDEX_NAME, RDB$INDEX_INACTIVE  
FROM RDB$INDICES WHERE RDB$INDEX_INACTIVE<>0;
```

and check the states of listed indices. Value 1 means that index is inactive, and value 3 means that index is non-active state.

To activate index, do the command for each index:

```
ALTER INDEX <indexnamexxx> ACTIVE;
```

## 4.2. Recalculate selectivity (statistics) of indices in your Firebird database

The optimizer in Firebird (and in InterBase) uses 2 main criteria when analyzing a query and choosing an execution plan—the size of the tables (cardinality) and the selectivity of the indices

(selectivity).

The selectivity of the indices is calculated as 1 divided by the number of unique values in the indexed column (or columns). For example, if this is an index on a primary key, and there are 10 thousand records in the table, then its selectivity will be equal to  $1 / 10000 = 0.0001$ . And if the same table has a column with 10 different values, then the index on this column will have a selectivity of  $1 / 10 = 0.1$ . For the optimizer, the lower the selectivity of the index, the better.

If the optimizer always determines the size of the table more or less accurately, then the selectivity of the indices is not recalculated automatically. As a result, the optimizer could estimate the selectivity that is far from reality. So, the selectivity needs to be recalculated manually.

The recalculation of the selectivity of the indices occurs when they are created, when **ALTER INDEX ACTIVE**, and when **SET STATISTICS INDEX indexname**.

For example, you created a table, created indices, and then filled the table with data. The selectivity of all the indices of this table will remain 0 until you recalculate the selectivity.

Or, you created a table, indices, filled it with data, made a backup-restore (when restoring, the indices are recreated, and the selectivity will be relevant). And then you changed 30-40% of the data in some tables (added, deleted, or modified). And the selectivity will also be irrelevant.

You can view the stored selectivity of the indices with a query (the selectivity of inactive indices will be equal to -1)

```
select i.rdb$relation_name, i.rdb$index_name, i.rdb$statistics
from rdb$indices i
-- do not show indices for system tables
-- where coalesce (rdb$system_flag, 0) = 0
order by 1, 2
```

Recalculating statistics of indices in Firebird database is a way to improve the performance of your queries. Statistics tell Firebird how selective your indices are, and help it choose the best execution plan. Here are the steps to recalculate statistics of indices in Firebird database:

- Connect to your database using a tool like isql, FlameRobin, etc.
- Run the command **SET STATISTICS INDEX <index\_name>;** for each index you want to update. You can also use **ALL** instead of **<index\_name>** to update all indices in the database.
- Commit your changes with the command **COMMIT;**
- You can check the updated statistics with the command **SHOW INDEX <index\_name>;** or **SHOW INDEX ALL;**

That's it. You have recalculated statistics of indices in Firebird database. You can do this periodically, especially after large changes in your data, to keep your indices efficient.

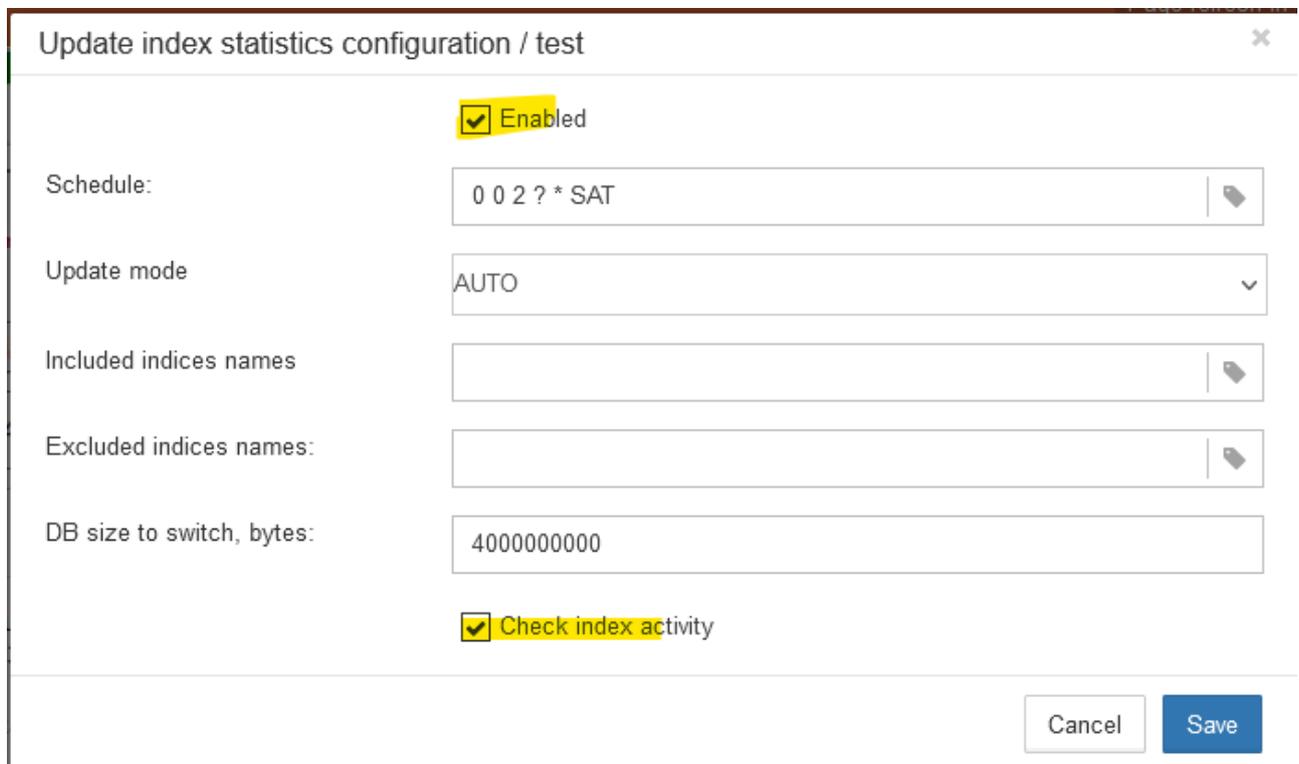
## 4.3. How to check indices state and recalculate statistics with HQbird

To verify the state of indices with HQbird, follow these steps:

- Access the web interface at <http://localhost:8082> and choose the “Index statistics recalculation” job for the registered database. This job will do both tasks.



- Open the dialog and enable the job.



- The job will perform both statistics recalculation and index state check. The default schedule is weekly on Saturdays at 02:00 am.

If you need to do immediate recalculation of indices statistics (for example, if you just installed HQbird), you can click “Run Now” in the job’s widget.

- For databases larger than 20 GB, we suggest recalculating indices monthly instead of weekly.

# Chapter 5. Step 5 of 5. Choose proper page size for your Firebird database

## 5.1. A bit of theory

Your Firebird database is made up of pieces that are all the same size. These pieces are called pages. The size of the pages depends on the version of Firebird you use. In Firebird 1, the default size of pages was 1024 bytes. In Firebird 2.5, the default page size was 4096 bytes. And in Firebird 3.0 and up, the databases are created with page size 8192 bytes. For bigger databases, Firebird 2.5 and 3 have options up to 16384, and v4 + has 32768 bytes.

	1024	2048	4096	8192	16384	32768
2.5	possible	possible	default	possible	possible	Not implemented
3.0	Not supported	Not supported	possible	default	possible	Not implemented
4.0	Not supported	Not supported	possible	default	possible	possible
5.0	Not supported	Not supported	possible	default	possible	possible

To see what is the page size for your database, please use command:

On Windows

```
gstat.exe -h Disk:path\database.fdb
```

On Linux

```
/opt/firebird/bin/gstat -h /pathto/database.fdb
```

and see line “Page size”.

The size of the pages matters a lot for your Firebird database, so you need to pick the right one.

To change page size, you need to use the **gbak** tool to backup and restore your database, and add the **-page <NNN>** option. But be careful, this can take a long time if your database is big. So, it's better to pick the right page size from the start.

How do you pick the right page size for your Firebird database? There are the following things to consider:

1. Default page size
2. Database size
3. Depth of indices
4. Size of compressed records stored on pages for the large tables
5. To do — recalculate configuration after change of page size

Let's consider all these points:

## 5.2. Default page size

As you can understand, the page size cannot be less than default on the specific database. If database is created with `CREATE DATABASE` command without `PAGE_SIZE` option, the database by default will be created with default page size. If database was created on previous version of Firebird and converted with backup/restore, the page size could be preserved, and it could be less than default, i.e., non-optimal. In this case, you need to change page size to the default or above. Practically, it means 8192 as a minimum.

## 5.3. Database size

For databases less than 20Gb in size, or without intention to grow over this size without backup/restore (when there will be a possibility to change the page size), the general recommendation is to use 8192 bytes as page size (and it is a default size for versions 3+).

For bigger databases use 16384 bytes.

What about 32768, will be better to use the biggest available option?

We have made tests with databases up to 1Tb in size, and did not find the direct performance improvement in comparison with 16384. One of our customers with 4Tb database has reported that 32768 page improved performance. So, decision to use 16K or 32K page size will depend on the next steps.

## 5.4. Depth of indices

In order to keep performance of indices on the optimal level, they should have `depth` `←3`.

How to check depth?

Using only standard tool, it is a bit lengthy process:

1. Run `gstat` tool to collect statistics

```
gstat -r -user SYSDBA -pass masterkey localhost:Disk:pathdb.fdb > stats.txt
```

2. Open statistics in text editor and find all occurrences of words "Depth: 3", "Depth: 4", "Depth: 5". There should be indices with depth 2 and 3 (3, if database is not very small), and, for optimal

performance, should be no indices with Depth 4 and 5.

In HQbird, you need to use tool **HQbird Database Analyst** from HQbird Admin package, which is designed to analyze databases statistics.

Open HQbird Database Analyst, retrieve database statistics (or open file with manually collected statistics), go to tab Indices, and sort on Depth:

Index	Table	Depth	Keys	Key Len	Max Dup	Total Dup	Uniques	Size, ...	AvgFill	Ratio %
PK_CAIXAM	CAIXAM	3	25515	6.11	0	0	125515	1.46	98	0
PK_PAF_FV	PAF_FV	3	85829	2.31	1	10	285819	2.10	100	0
PK_PAF_FV_PAGTO	PAF_FV_PAGTO	3	36432	1.01	0	0	336432	1.85	100	0
FK_PAGAMENTO2	PAGAMENTO	3	28727	0.05	66634	226921	1806	1.55	76	0
PK_PAGAMENTO	PAGAMENTO	3	28727	20.63	0	0	228727	5.98	99	0
PK_PAGAMENTO_REGISTRO_ITEM	PAGAMENTO_REGISTRO_ITEM	3	93280	28.22	0	0	93280	3.05	100	0
FK_PARCELAMENTO_DEBITOS_2	PARCELAMENTO_DEBITOS	3	43254	23.51	2	717	42537	1.32	94	0
PK_PARCELAMENTO_DEBITOS	PARCELAMENTO_DEBITOS	3	43254	25.65	0	0	43254	1.31	100	0
PK_PRODUTO_DESCRICAO	PRODUTO_DESCRICAO	3	53360	1.00	0	0	453360	2.54	99	0
PRODUTO_DESCRICAO_IDX1	PRODUTO_DESCRICAO	3	53360	0.33	27052	382019	71341	3.60	72	0
FK_PRODUTO_HPRECO	PRODUTO_HPRECO	3	06508	0.31	383	378143	28365	3.32	69	0
PK_PRODUTO_HPRECO	PRODUTO_HPRECO	3	06508	1.60	0	0	406508	3.99	73	0
PK_VENDA	VENDA	3	22268	9.03	0	0	122268	1.81	98	0
FK_VENDA_ITEM1	VENDA_ITEM	3	85276	3.88	54	163008	122268	2.62	98	0
FK_VENDA_ITEM2	VENDA_ITEM	3	85276	0.07	1246	270035	15241	2.64	61	0
PK_VENDA_ITEM	VENDA_ITEM	3	85276	6.61	0	0	285276	3.40	99	0
PK_VENDA_FV	VENDA_FV	3	42330	9.14	0	0	142330	1.96	100	0
PK_ABC_TEMP	ABC_TEMP	2	1000	1.05	0	0	1000	0.01	60	0
FK_ALUGUEL1	ALUGUEL	2	4854	0.01	1236	4840	14	0.04	62	0

If you see indices with **Depth > 3**, it is necessary to increase database page size.

If you already using the maximum possible page size, for example 16384 on v3, consider migration to v4-5, where maximum page size is 32768.

## 5.5. Size of compressed records

Firebird keeps table's data in the compressed form. You can see in the statistics of databases from the previous step an information about size of records in the table, something like this: (the output can be different for different versions):

TST (128)

Primary pointer page: 148, Index root page: 149

Average record length: 20.99, total records: 100000

Average version length: 0.00, total versions: 0, max versions: 0

Data pages: 1516, data page slots: 1516, average fill: 62%

Fill distribution:

0 - 19% = 1

20 - 39% = 0

40 - 59% = 3

60 - 79% = 1512

80 - 99% = 0

It is necessary to check the size of records for the biggest tables and compare it with the page size.

The idea is to calculate how many records are stored on the single page — if there are records with average length 700 bytes on 4096 bytes pages, Firebird will need to read many data pages.

Of course, it depends on the database structure and the business logic, but the bigger size of records, the bigger page size needed.

In “HQbird Database Analyst” we can see records sizes for tables in the simple table form: open or retrieve statistics, tab “Tables”, and sort on **Record Length**.

In addition to “Record Length”, “Database Analyst” show parameter “Average Fill” and calculates derived parameter “Real Fill”, to show how efficiently data are stored on database pages.

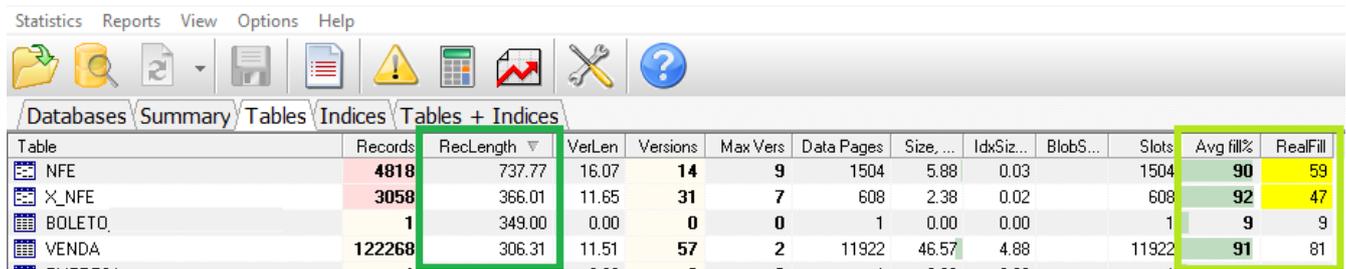


Table	Records	RecLength	VerLen	Versions	Max Vers	Data Pages	Size, ...	IdxSiz...	BlobS...	Slots	Avg fill%	RealFill
NFE	4818	737.77	16.07	14	9	1504	5.88	0.03		1504	90	59
X_NFE	3058	366.01	11.65	31	7	608	2.38	0.02		608	92	47
BOLETO	1	349.00	0.00	0	0	1	0.00	0.00		1	9	9
VENDA	122268	306.31	11.51	57	2	11922	46.57	4.88		11922	91	81

## 5.6. Recalculate configuration after change of page size

Remember that Firebird cache size is measured in pages? If page size is changed, the absolute size of cache will be also changed. If you have set 1 million of pages for the page size 8192, the absolute size of the cache is 8Gb, and if page size is increased to 16384 bytes, the absolute cache size will become 16Gb.

In order to avoid problems with too big cache, after the change of page size, recalculate the configuration in Configuration Calculator [cc.ib-aid.com](http://cc.ib-aid.com).

# Chapter 6. Summary

While the above steps are crucial for enhancing performance, they may not be comprehensive. Should you find that, despite thorough checks, the performance remains poor, it may be necessary to refine your queries, streamline your architecture, and fine-tune your transactions.

For expert assistance in performance optimization, feel free to engage our optimization services by reaching out to us at [support@ib-aid.com](mailto:support@ib-aid.com).

We offer two primary service options, both of which can be customized to suit your needs:

- **Incident Resolution:** This includes two remote desktop sessions, priced at USD\$499, designed to address and resolve issues swiftly.
- **Comprehensive Optimization Service:** Priced at USD\$2990, this service entails a thorough optimization process, including monitoring of production environments and multiple iterations for best results.