

**IBSurgeon**

# FBDataGuard 3.0 User Guide

Version 1.6.1

## Table of Contents

FBDataGuard for Windows User Guide.....	4
1. What is IBSurgeon’s FBDataGuard? .....	4
About IBSurgeon .....	4
2. Installation.....	4
2.1. Download and registration.....	4
2.2. Installation on Windows.....	6
2.3. Choosing folders to store configuration, backups and statistics .....	7
2.4. Installation on Linux .....	8
3. Initial FBDataGuard Configuration .....	10
3.1. Launch web-console.....	10
3.2. Auto discovery feature of FBDataGuard .....	11
3.3. Firebird server registration .....	12
3.4. Firebird database registration.....	14
3.5. Email alerts in FBDataGuard.....	15
3.6. Next steps with FBDataGuard .....	17
3.7. Embedding FBDataGuard into your own application.....	17
4. Configuring FBDataGuard with Web-console .....	18
4.1. Overview of Web-console .....	18
4.2. Server: General configuration .....	19
4.3. Server: auto updates .....	20
4.4. Server: Agent Space .....	21
4.5. Server: Server version .....	21
4.6. Server: Server log .....	22
4.7. Server: Temp files.....	22
4.8. Server: Server space.....	23
4.9. Server: Send logs .....	24
4.10. Database: General configuration .....	24
4.11. Database: Transactions .....	25
4.12. Database: Index statistics.....	25
4.13. Database: Active users .....	26
4.14. Database: Backup.....	27
4.15. Database: Store metadata .....	29

4.16. Database: Validate DB.....	30
4.17. Database: Delta-files monitoring .....	30
4.18. Database: Disk space.....	31
4.19. Database: Database statistics .....	31
4.20. Database: Send logs .....	32
5. FBDataGuard tips&tricks .....	32
5.1. Path to FBDataGuard configuration .....	32
5.2. Adjusting web-console port .....	33
Appendix: CRON Expressions .....	34
CRON Format.....	34
Special characters.....	34
CRON Examples .....	35
Notes .....	36
6. Support contacts .....	36

## FBDataGuard for Windows User Guide

### 1. What is IBSurgeon's FBDataGuard?

FBDataGuard is a server-side tool to monitor Firebird databases and prevent corruptions. It watches for databases and server environment, performs backups in right way, gathers statistics and sends alerts about actual and possible problems.

FBDataGuard is intended to be automatic maintainer and administrator assistant for important Firebird databases, especially at remote locations and being bundled with Firebird-based software.

#### About IBSurgeon

IBSurgeon ([www.ib-aid.com](http://www.ib-aid.com)) was founded in 2002 with the idea to provide InterBase and Firebird developers and administrator with unique services and tools focused on databases safety, performance and availability. IBSurgeon is a Platinum sponsor of Firebird Foundation and, as a member of Technical Task Group, has strong relationship with Firebird Project, with direct representatives in Firebird-Admins and in Firebird Foundation Committee. Today, IBSurgeon serves more than 2000 companies worldwide with emergency, optimization and maintenance tools and various services.



## 2. Installation

### 2.1. Download and registration

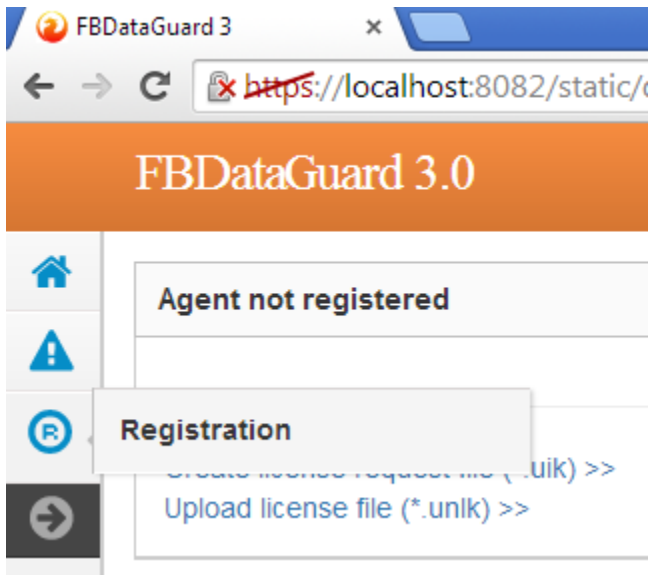
Every instance of FBDataGuard should be registered according its license agreement (only exception is special bundle for ISV, designed to be installed and used as part of third-party application, contact [isv@ib-aid.com](mailto:isv@ib-aid.com) for details).

#### How to download FBDataGuard

- 1) Get account for IBSurgeon Deploy Center - i.e., username and password to download and activate IBSurgeon products. Visit <http://ib-aid.com/en/fbdataguard/> for more details.
- 2) Login to IBSurgeon Deploy Center <http://deploy.ib-aid.com> with your username and password and verify that you have specified correct email – it will be used to send you license information.
- 3) Choose FBDataGuard in the list of available products' licenses and choose link "Download", then **download FBDataGuard**, unpack it using password near file name and install.

#### How to register FBDataGuard

- 1) After installation of FBDataGuard (see [2.2. Installation on Windows](#)) and adding Firebird server to monitoring click on "Registration" tab at FBDataGuard:



- 2) Choose link [Create license request file \(\\*.uik\) >>](#)
- 3) Enter you name, company and email. Important! Email should be the same as specified in IBSurgeon Deploy Center account, otherwise registration will fail:

Create license request x

---

Name:

Company:

E-mail:

- 4) After it save generated file license.uik to your computer
- 5) Login to IBSurgeon Deploy center <http://deploy.ib-aid.com>, and choose link "Activate"



List of owned products' licenses:

Product name	Description	Downloads	Available activations	Used activations	End license date	Actions
FBDataGuard 2.5	Monitor, recover, protect Firebird database	<a href="#">Download</a>	2	0	2099-12-29	<a href="#">Activate</a>

- 6) Upload file license.uik to IBSurgeon Deploy Center

Available registrations	Log off	Personal Info	
-------------------------	---------	---------------	--

Create new (1) or use existing activations (2) from list below

1. Create new activation

To activate your application, please upload UIK file, generated by your application's registration wizard

**Please select file with your registration information (UIK):**

Description (for your purposes)

UIK file name:

- 7) Download unlock file from the list of available activations and save on your computer. Also it will be sent to you by email.
- 8) Go back FBDataGuard, tab "Registration" and click [Upload license file \(\\*.unlk\) >>](#)
- 9) Choose fbdg.unlk and FBDataGuard on this computer will be registered.

### Note:

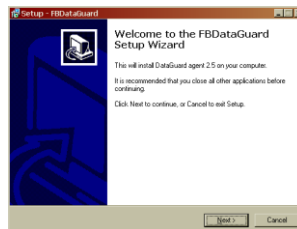
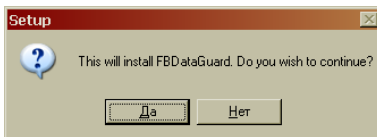
FBDataGuard registration is based on **AgentID**, **network name** and **IP address** of the server where FBDataGuard is installed. If they will be changed, FBDataGuard will turn into unregistered state.

## 2.2. Installation on Windows

In this section we discuss installation process of FBDataGuard for Windows.

*FBDataGuard should be installed on the same computer where Firebird is running. It cannot monitor Firebird server, databases and hardware environment from the remote computer.*

To install FBDataGuard you need to start installer with administrators' rights and go through several steps. First steps are obvious: start, notification of what will be installed and accepting license agreement.



Then you need to select folder where to install FBDataGuard. By default it offers **C:\Program Files\FBDataGuard** location. You can accept it or choose another location.

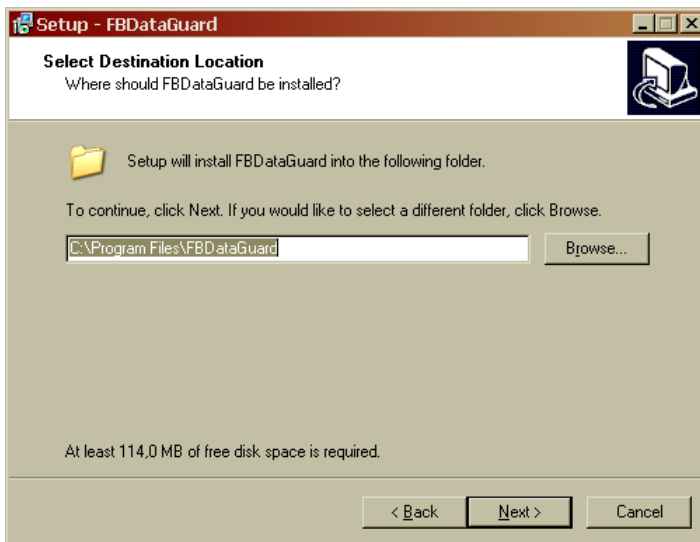


Figure 1 Select where to install FBDataGuard

### 2.3. Choosing folders to store configuration, backups and statistics

Next step is important. FBDataGuard installer needs to know places where to store FBDataGuard configuration and backups, statistics.

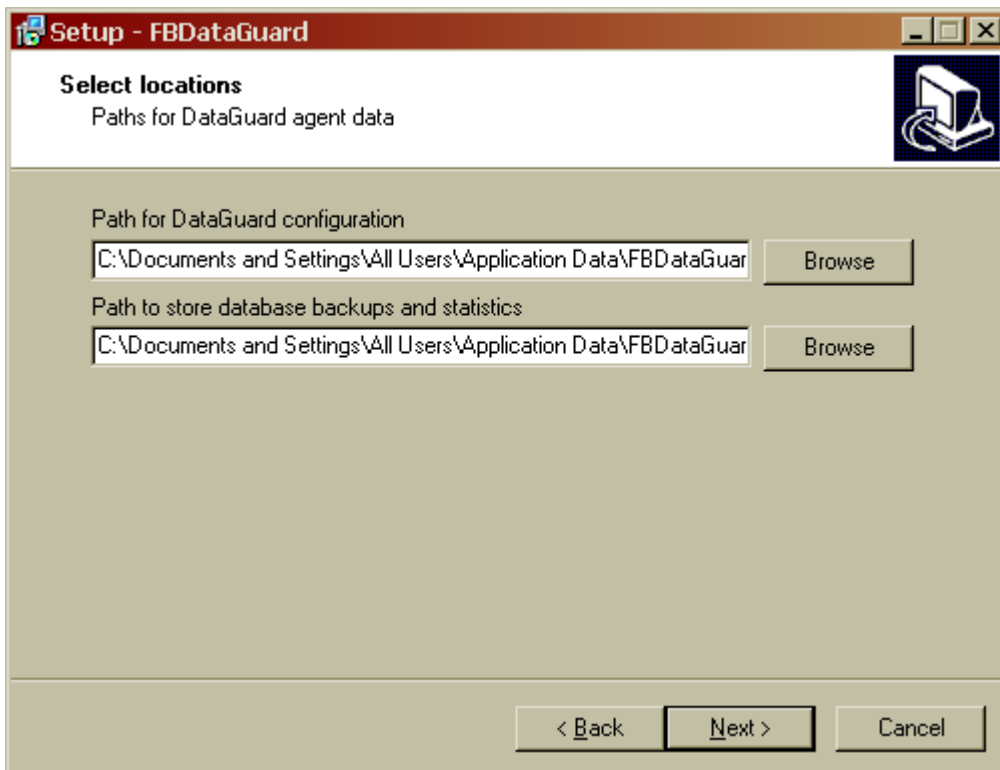


Figure 2 Choose folders to store FBDataGuard configuration and backups/statistics

We strongly recommend **do not use default** settings. The best way is to point configuration and backup folders to the **local drive** with backups (not mapped). For example, if your database is placed at D:\, and **backups is at F:\**, we would recommend to use something like this:

Path to DataGuard configuration

**F:\FBDataGuard\config**

Path to store database backups and statistics

**F:\mybackups**

*You can also explicitly specify path to the folder with backup later, during configuration of monitoring server, but we recommend doing it during installation. Folder to store backups usually requires a lot of space.*

After that you will be requested to confirm that you wish to install FBDataGuard as a service and start this service. Proceed with other steps by clicking “Next” and finish installation. Then we need to proceed with FBDataGuard initial configuration.

## 2.4. Installation on Linux

There is no Linux (as well as Mac OS X) installer in FBDataGuard 2.8 You need to download archive dataguard30.zip ([www.ib-aid.com/dgupdates/dataguard30.zip](http://www.ib-aid.com/dgupdates/dataguard30.zip)) and follow these instructions:

### Prerequisites

For all Linux versions:

- 1) Unzip this package to the suitable folder (for example, /opt/dataguard30)
- 2) Change owner of folder dataguard30 to user *firebird*:  
**chown -R firebird dataguard30**
- 3) add run permissions for the necessary files:

```
run.sh (all Linux versions)
dg_debian_run.sh (Debian)
dg_centos.sh (CentOS)
dg_suse.sh (Suse)
```

- 4) Install Java. It could be Sun (Oracle) Java or OpenJDK 1.7 and higher (however, it will work with 1.6 too, with some limitations).

More information: <http://openjdk.java.net/install/>

Short commands for popular distributives:

Ubuntu

```
$ sudo apt-get install openjdk-7-jre
```

OpenSuse

```
$ sudo zypper install java-1.7.0-openjdk
```

Fedora, Red Hat Enterprise Linux, CentOS

```
$ su -c "yum install java-1.7.0-openjdk"
```

(or look for specific OS instructions)



Debian

```
$ sudo apt-get install openjdk-7-jre
```

5) Try to run FBDataGuard as an application (from /opt/dataguard30):

```
./run.sh
```

If Java was installed correctly, FBDataGuard should start as application. Check the output and close it.

### *Steps to install*

Next steps are required to run FBDataGuard as a service. They require superuser rights (**sudo** or **su**).

#### **Debian**

1) install jsvc service runner (java will be installed as dependency):

```
$ sudo apt-get install jsvc
```

2) Check **JAVA\_HOME** location and if it differs from /usr/lib/jvm/java-7-openjdk/jre, set correct location (JAVA\_HOME=) in the file **dg\_debian\_run.sh**

3) Set actual location of FBDataGuard to **DG\_HOME** (by default it's /opt/dataguard30)

4) Start FBDataGuard from FBDataGuard folder

```
./dg_debian_run.sh start
```

5) By default FBDataGuard *listens to localhost* - i.e., it's accessible *only locally* (http://localhost:8082). To enable external connections, go to /opt/dataguard30/conf/network.properties and set **server.bind-address** to the actual IP value (type ifconfig to find out your IP, /sbin/ifconfig at CentOS)

#### **CENTOS**

1) Install service with command

```
./dg_centos.sh install
```

2) Run FBdataGuard service with command

```
./etc/init.d/dataguard30 start
```

#### **OpenSuse**

1) Install service with command

```
./dg_suse.sh install
```

2) Run service

```
./etc/init.d/ dataguard30 start
```

For other Linux distributives – use **dg\_centos.sh**.

### 3. Initial FBDataGuard Configuration

After installing FBDataGuard we need to configure it. Please follow these steps:

1. Make sure that you have Firebird 1.5 or later, and it is working;
2. FBDataGuard service is installed and started properly. You can check it using Services applet in Control Panel (right-click on "My Computer", choose "Manage", then "Services and Applications", "Services" and find in the list "FBDataGuard Agent"
3. Make sure the FBDataGuard port is accessible (8082) and it is not blocked by firewall or any other antivirus tools. If necessary, adjust port in FBDataGuard configuration (see [5.2. Adjusting web-console port](#)).

**Important:** On Windows systems without installed Microsoft Visual Studio redistributable package the right elevation software (RunAsAdmin.exe) will not work, so you need to run install and uninstall scripts (procrun-install and procrun-uninstall) manually from command prompt launched with as Administrator privileges.

#### 3.1. Launch web-console

To open web-console type in your browser <https://localhost:8082> or <https://127.0.0.1:8082>

Or you can choose in "Start" menu FBDataGuard\ "Launch the DataGuard web console for localhost"

#### Supported browsers

FBDataGuard web-console should work correctly with Firefox, Safari and Internet Explorer.

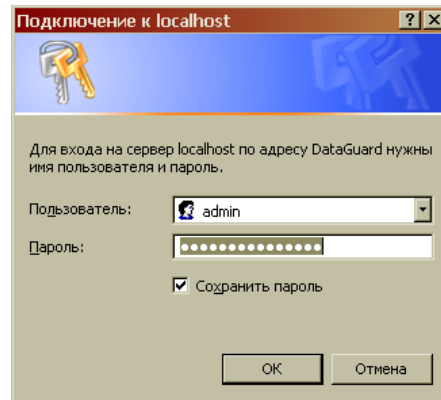
#### Error message regarding webs-site certificate

Initially browser will indicate that this web-site (<https://localhost:8082>) is not safe, and it will recommend leaving web-site. This message is caused by the default security certificate for FBDataGuard web-console.

Please ignore this message and click to open FBDataGuard web-console.

It will ask you for username and password (login dialog can be different for Firefox, Safari or Chrome)

Default username/password for web-console is "admin"/"strong password" (no quotes).



**Figure 3** Enter username and password for FBDataGuard web-console

### 3.2. Auto discovery feature of FBDataGuard

At first launch FBDataGuard will check computer for installed Firebird servers. FBDataGuard for Windows search registry for Firebird records, FBDataGuard for Linux and Mac OS X checks default location of Firebird installations.

FBDataGuard will show the list of all Firebird copies installed, but only the one instance of Firebird can be monitored by FBDataGuard. Choose it by clicking **“Add to monitoring >>”**

If you don't see Firebird instance in auto discovery list, you can choose **“Add custom >>”** and configure instance parameters manually.

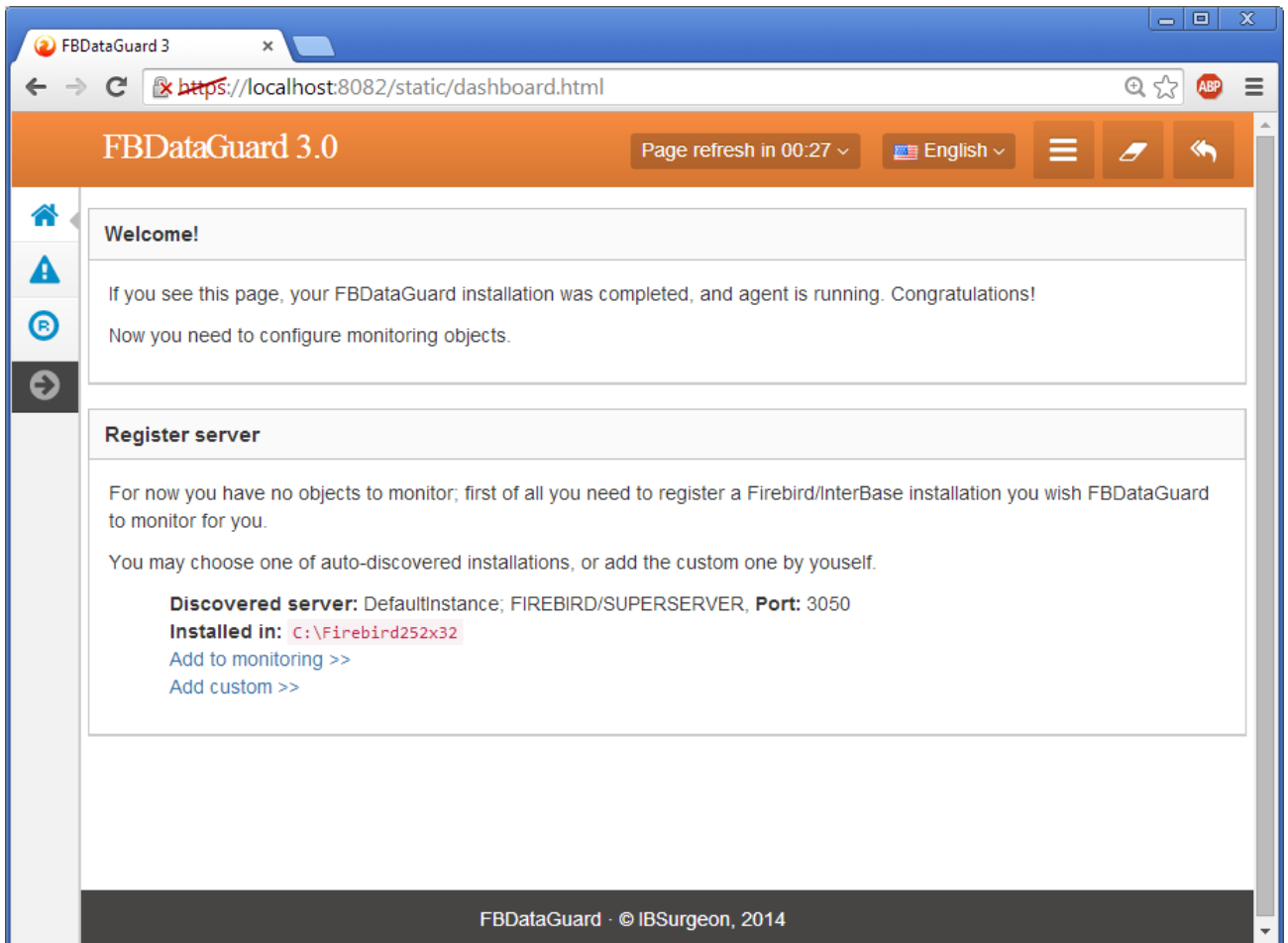


Figure 4 Auto discovery in FBDataGuard

### 3.3. Firebird server registration

To register auto-discovered server, you need to click at “Add to monitoring>>” and then adjust auto-discovered settings.

*Note: to use Windows Trusted Authentication (by default it's off) you need to be sure that libraries jaybird21.dll and fbclient.dll (from appropriate Firebird version) are in searchable Windows paths.*

FBDataGuard offers default values for Firebird server. We strongly recommend changing of “Output directory” parameter.

The screenshot shows the 'Register Firebird server' dialog box. The fields are as follows:

- Installed in: C:\Firebird252x32
- Host: localhost
- Port: 3050
- Use trusted auth:
- SYSDBA login: SYSDBA
- SYSDBA password: masterkey
- Output directory: \$\{agent.default-directory}\\$\{server.id}

Buttons: Cancel, Save

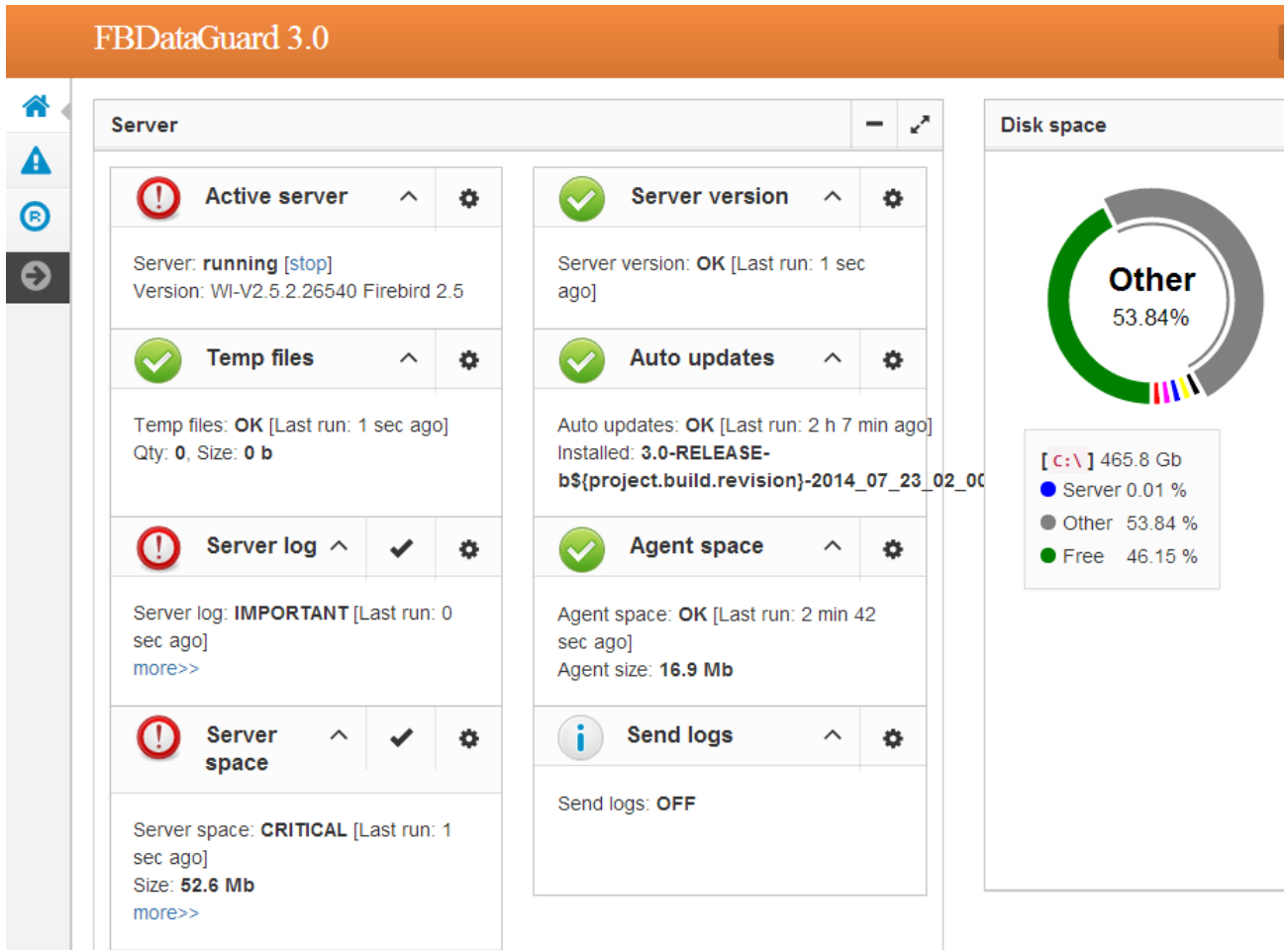
**Figure 5 Register server in FBDataGuard**

By default “Output directory” for Firebird server is **`$$\{agent.default-directory}\$\{server.id}`**

It means that backups, statistics and gathered logs will be stored in the folder for backups specified in [\[2.3. Choosing folders to store configuration, backups and statistics\]](#).

It can be not very convenient, so we recommend pointing FBDataGuard output directory to more simple path, usually located at disk where backups are intended to be stored, for example **F:\myserverdata**

After clicking “Register” FBDataGuard will populate default configurations files and immediately start analysis of firebird.log. It can take a while (for example, 1 minute for 60Mb firebird.log). After that you will see initial web-console with registered Firebird server:



**Figure 6** Web-console with registered Firebird server

FBDataGuard shows alerts and statuses of monitored objects. As you can see, in this example FBDataGuard found error messages in firebird .log and determined that size of Firebird installation server is too big.

In [\[Configuring FBDataGuard\]](#) we will consider in details each monitored objects and its settings.

**Note:** you cannot delete registered Firebird server in FBDataGuard 2.8 web-console of Control Center. The only way to unregister server is to delete its configuration files. In general, there is no reason for deleting registered server, until you want completely uninstall FBDataGuard.

Now we need to proceed with database registration.

### 3.4. Firebird database registration

To register database in FBDataGuard, you need to click at “Add database to monitoring>>” and fill the following form:

#### Add database for guard

---

Database name:

DB alias:

Path to database:

Output directory:

---

*i* Loading... ↻
Cancel Save

At the form header you’ll see database GUID – it’s used for unique identification of database. It is used only internally, there is no need to remember it.

“**Database name**” is for convenience while referring database in alerts and email messages.

“**DB alias**” is a database alias from aliases.conf. If you specify both “DB Alias” and “Path to database”, “DB Alias” will be used.

“**Path to database**” is the local path to database (remember that FBDataGuard should work at the same computer with Firebird). If you are putting database on external drive, it can raise error “File... has unknown partition”. To fix it you need to click on “Configure” at Server widget and click “Save” to make FBDataGuard re-read partitions.

“**Output directory**” is the folder where FBDataGuard will store backups, logs and statistics for this database. It’s a good idea to specify “Output directory” to some explicit location like F:\mydatabasedata

Note: you can specify exact absolute locations for backups and statistics later in appropriate dialogs (see [[Configuring FBDataGuard](#)]).

After registration FBDataGuard will populate database configuration with default values and then show web-console with registered database:

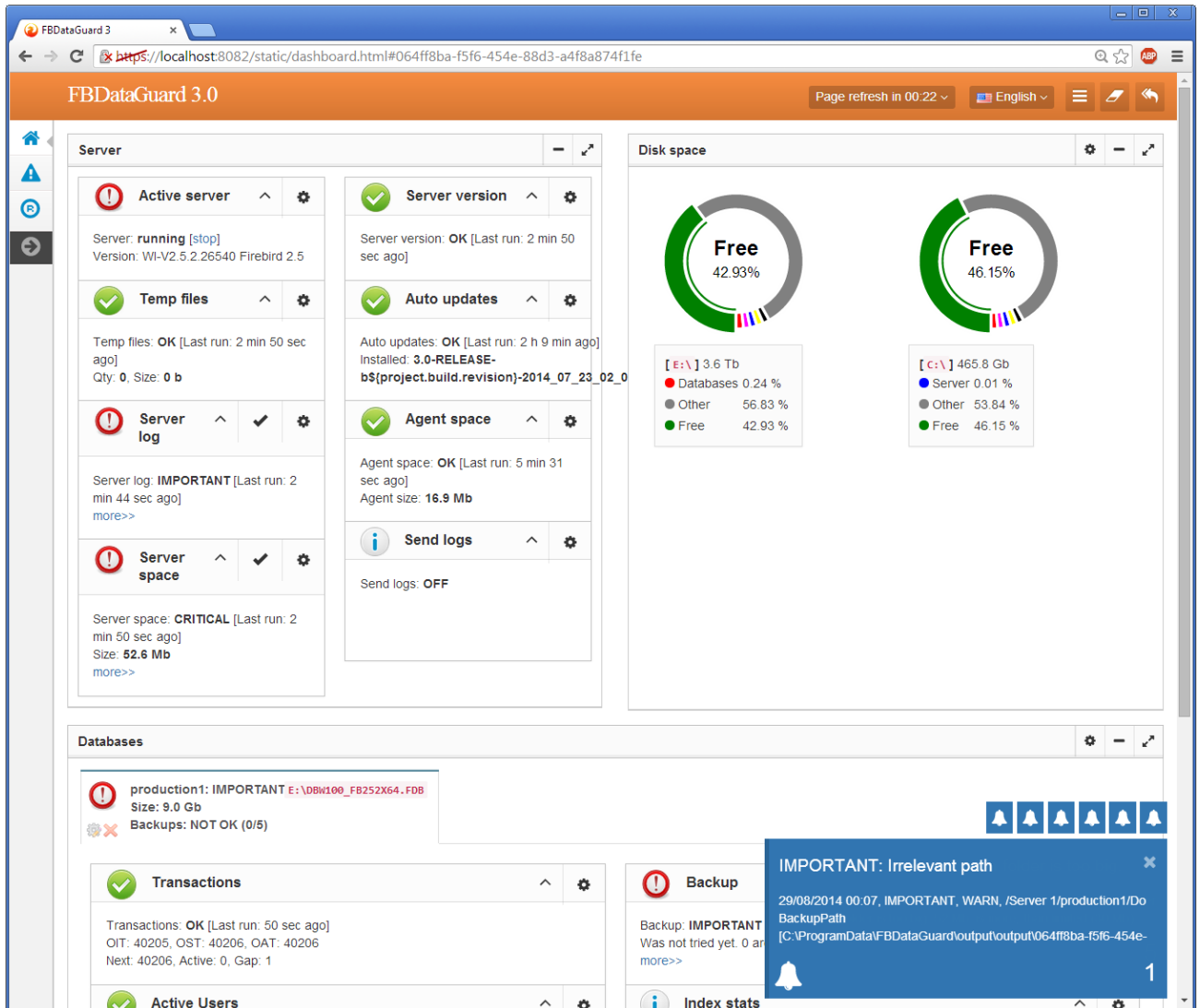


Figure 7 FBDataGuard web-console with added database

You can adjust database settings later; now let's proceed with alerts setup.

### 3.5. Email alerts in FBDataGuard

FBDataGuard can email alerts to administrators which contain information about successful backups and potential and real troubles with databases.

It's a very good idea to setup emailing of alerts. To do this you need to switch to tab "Alerts" and then click to sign "Configure":

00:43 ▾ English ▾ ☰ ✎ ↶

✎ ⚙ - ↶

Copy Print Save **configure** ▾

**FBDataGuard 3.0**

Home Server

Alerts

---

Alerts configuration ✕

---

Installation name:

Myagent

Installation GUID:

VJQ-K><BVSQ-TURG

Raise warning for suspicious backup paths:

Sent alerts by e-mail:

Anti-spam delay, minutes:

60

Send alerts to:

youremail@company.com

'From' field:

youremail@company.com

SMTP server address:

smtp.company.com

SMTP server port:

25

Secure (SSL) connection:

SMTP server login:

support

SMTP server password:

your-password-comes-here

Cancel Save

**Figure 8** Email alerts configuration dialog in FBDataGuard



First of all, you need to enable alerts sending by enabling checkbox **“Send alerts by e-mail”**.

**“Installation name”** is human readable name for your convenience; it will be referred in emails and alerts.

**“Installation GUID”** is a service field; there is no need to change it.

**“Raise warning for suspicious backup paths”** is to warn about potentially wrong backup file, for example, if database is stored in “Documents and Settings” or the length of backup path is too long.

**“Anti-spam delay”** specifies delay on sending repeating messages. It avoids flooding of administrators' mailbox with repetitive messages. 60 minutes is an optimal value.

**“Send alerts to”** specifies where to send emails.

**“From field”** is what will be set as sender in the email.

**“SMTP server address”**, **“SMTP server port”**, **“SMTP server login”** and **“SMTP server password”** are data which will be used to send emails.

Click **“Save”** to save email alerts settings. *There can be delay while saving, since FBDataGuard is checking the settings and send test email to the specified address.*

### 3.6. Next steps with FBDataGuard

After you have setup FBDataGuard and added there server and database(s) to be monitored, you need to adjust settings for the most important maintenance activities: Backup and Space Monitoring. Other activities can be configured later.

Please now proceed to sections [4.14. Database: Backup](#), [4.8. Server: Server space](#) and [4.18. Database: Disk space](#) to get more information.

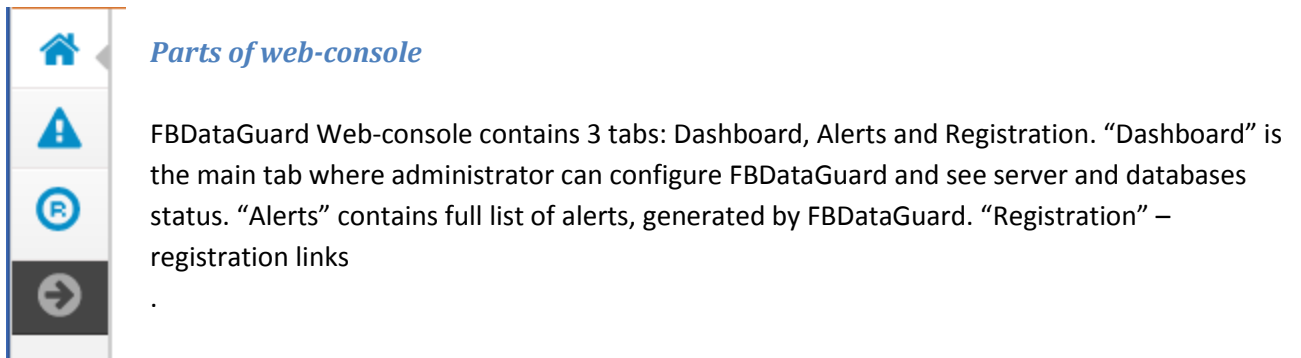
### 3.7. Embedding FBDataGuard into your own application

It's easy to embed FBDataGuard into your own application, so it will be installed silently and it will watch for your database.

If you are a vendor of Firebird-based database application and you need to know how to embed FBDataGuard into the installation package of your application (please note that special license is required for bundling FBDataGuard with your applications) please contact IBSurgeon at [dg@ib-aid.com](mailto:dg@ib-aid.com) and we will provide you with all necessary information.

## 4. Configuring FBDataGuard

### 4.1. Overview of web-console



#### Jobs

FBDataGuard web-console is intended for easy editing of configurations of activities (called “**jobs**”) which are fulfilled by FBDataGuard (not all jobs are listed in the web-console, some of them are available only through direct editing of configuration files).

Almost all FBDataGuard jobs have 2 purposes: the first is to monitor for some value and raise alerts if necessary, and the second is to store historical values into logs, so later it’s possible to see the dynamics of all parameters of Firebird server and database.

In this section we will consider general configuration of jobs parameters, but not an analysis of gathered log.

#### Jobs widgets

General approach is the following: each activity is represented by a “widget”, which has the following parts:

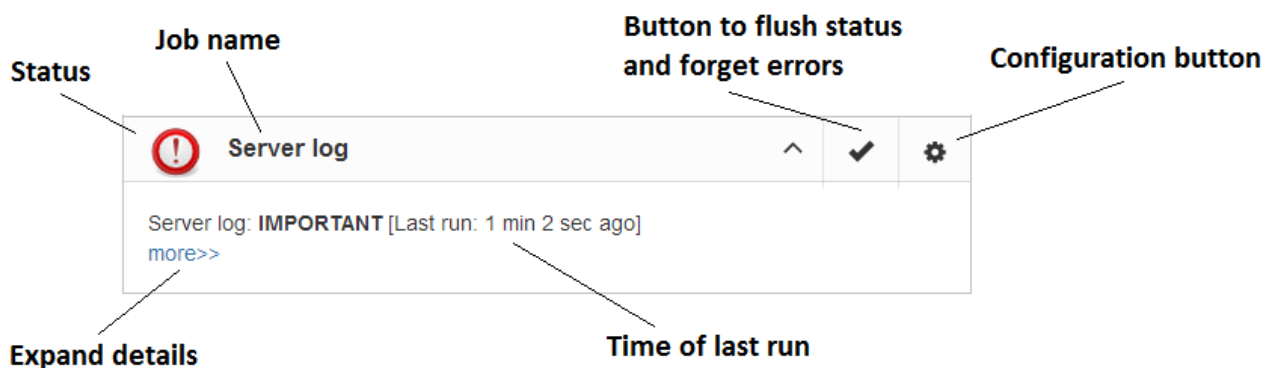


Figure 9 Elements of FBDataGuard web-console widget

**Status** – it is indicated with color icon and name. Status of database is a summary of all included server-level jobs and databases’ statuses, and, respectively, status of database is a summary of all database-level jobs.

#### Status types

**CRITICAL** means problems, **OK** means “Everything is fine”, **WARNING** means some issues which require attention, **MAJOR** means major issue, **MINOR** – minor issue, **MALFUNCTION** means that jobs

was not succeeded (something prevents its execution), **NOT\_AVAILABLE** means that job is not available in this server or database version.

**OFF** means that job is not active, **UNKNOWN** means that job is active but was not started yet, so actual results are unknown.

**Job name** indicates the name of activity.

**Configuration link** opens configuration dialog, which is individual for each job.

**Resolved** is the link to flush the status to UNKNOWN and forgot errors which were discovered previously. The status will be updated according the current situation after the next execution of the job.

**Last run** shows the time after the last run of this job.

**More>>** is the link which opens the widget and shows more details and suggested action for administrator to resolve the situation.

Additional links can be associated with database



All jobs in FBDataGuard have default settings which are very close to recommended values for the 80% of Firebird installations, so after initial configuration server and database will be protected at pretty good level comparing with default installation, but anyway we recommend additional configuration and tuning for every job.

In the next sections we will consider each job and its configuration.

#### 4.2. Server: General configuration

Server: General configuration widget shows summarized status of all server-level jobs and statuses of monitored databases.

**Server: General configuration** also indicates Firebird currently running or not and provides link to Stop/Start Firebird service (and only service – it does not affect Firebird which is running as application).

	<b>Active server</b>	^	
Server: <b>running</b> [stop]			
Version: WI-V2.5.2.26539 Firebird 2.5			

*Important: please be careful while using Start/Stop functionality in FBDataGuard. You can easily shutdown Firebird server which performs important work.*

If we click on **configure** link we will see the same dialog that we have used to register Firebird instance in FBDataGuard, and now it can be used for changing Firebird instance properties:

Server monitoring configuration: 064ff8ba-f5f6-454e-88d3-a4f8a874f1fe ✕

Installed in:

Host:

Port:

Use trusted auth:

SYSDBA login:

SYSDBA password:

Output directory:

Cancel

Save

The long set of letters and numbers is unique GUID number of this monitored Firebird server.

### 4.3. Server: auto updates

Auto update is an important job which notifies you that newer version of FBDataGuard is available and offers to update the software. It provides an alert regarding available updates and appropriate download link. Default time to run this job is 22-00 everyday (for information).

At configuration dialog of auto-updates you can disable auto check or set another time for it.

For more information about time format please refer to [Appendix: CRON Expressions](#)

Agent updates monitoring ✕

Enabled

Schedule:

Cancel Save

In fact there is a small confusion here: auto update does not perform automatic update of software, it just checks for the new versions periodically.

#### 4.4. Server: Agent Space

Agent space monitoring is intended to watch space occupied by logs, stats, metadata repository and other data, gathered by FBDataGuard. For databases unattended for a long time (1-2 years) it is possible that FBDataGuard logs will occupy too much space and lack of space can lead to database outage. To prevent it for sure, Agent Space is watching for occupied space.

By default Server: Agent Space job is enabled.

Also, if someone has ignored recommendations to put backups' folders to the explicit locations, it is possible that database backup will be created inside Agent folder. In this case you'll see CRITICAL status immediately – FBDataGuard will recognize and warn you regarding wrong configuration.

And, this job is useful for bundles of FBDataGuard and third-party applications.

In the configuration dialog you can enable/disable this job, set check period (by default it is 10 minutes), and set thresholds for alerts.

Thresholds can be set in % of max size occupied by log or using the explicit size in bytes.

FBDataGuard checks both values and raises alert for the first threshold. If you wish to set % only, you need to set -1 as value to "Max occupied".

#### 4.5. Server: Server version

FBDataGuard checks the version of Firebird server. If the version is not in the list of supported servers it gives a warning.

It is very simple job though useful in some cases jobs: it will ensure that user of Firebird-based application is working with recommended version.

Current version of FBDataGuard supports 1.5.x-2.5.x

##### Agent space monitoring ×

Enabled

Check period, minutes:

[Enable job](#)

10

Max occupied, %:

10

Max occupied, bytes:

250000000

##### Supported server version ×

Enabled

Schedule:

0 0 07 ? \* MON-FRI

Supported:

1.5, 2.0, 2.1, 2.5

#### 4.6. Server: Server log

“Server log” is one of the most sophisticated jobs in FBDataGuard from the point of view of its features and implementations, but it’s as easy to configure as other jobs are.

This jobs periodically checks firebird.log and if it detects that file was changed, log analysis starts. Embedded analytic engine checks each entry in firebird.log and classify it into several categories with different levels of severity.

According the severity of messages status of job is assigned and appropriate alerts are generated.

Once administrator has reviewed errors and alerts (and performed necessary actions to solve the reason of error), he clicks on “**Resolved**” link and FBDataGuard will forgot old error messages in firebird.log.

At configuration of “Server log” you can enable/diasble this job and set the check period (in minutes).

Also this job watches for the size of firebird.log and if its size exceeds “Size to roll”, FBDataGuard will split firebird .log and rename it according to date pattern

Parameter “Last error messages to store” specifies how many the most recent error messages will be stored.

#### 4.7. Server: Temp files

“Server: Temp files” job is very useful to catch and solve performance and outage problems with Firebird database.

While performing SQL queries Firebird stores intermediate results for sorting and merging data flows in temporary files, which are allocated in specified TEMP locations.

The screenshot shows a window titled "Server log" with a red warning icon. The content displays an important error message: "Error(s) in server log: IBASE11 (Server) Tue Jul 15 18:21:36 2014". The error details are: "Database: B:\DATABASE\JULY14\JOERG\DATENBANK\_E.IB", "Index 11 is corrupt (missing entries) in table DEVICE (136)". Below the error, there is a note: "Errors can indicate performance problems or database corruption. They should be carefully checked by database administrator."

#### Server log monitoring

Enabled

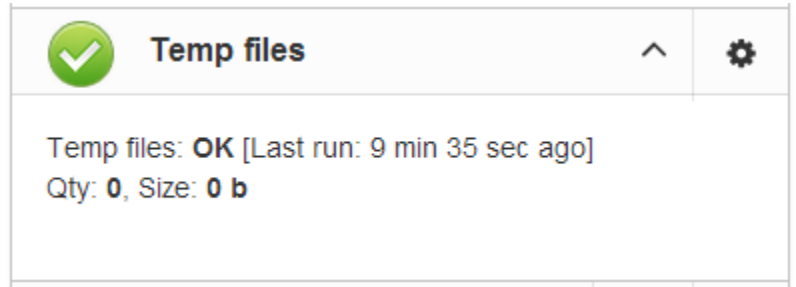
Check period, minutes:  Enable job

Size to roll, bytes:

Date pattern for rolling:

Last error messages to store:

FBDataGuard shows at “Server: Temp files” widget information about quantity and size of temporary files.



FBDataGuard recognizes locations of TEMP folders and monitors quantity and size of temporary files. Lack of space can lead to the performance problem or more serious errors, too many (or too large) temporary files can indicate problems with SQL queries quality.

Using configuration dialog you can enable/disable this job, set period and thresholds to the maximum size of temporary files (size of all files) and quantity.

Temporary files monitoring

Enabled

Check period, minutes:

10

Max size, bytes:

524288000

Max count:

1000

### 4.8. Server: Server space

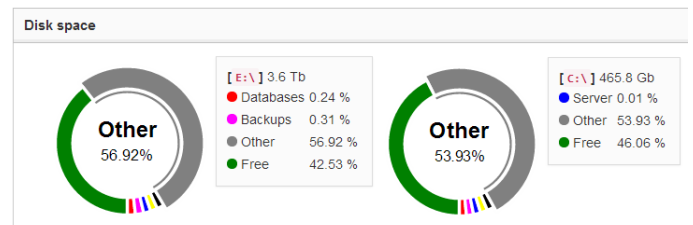
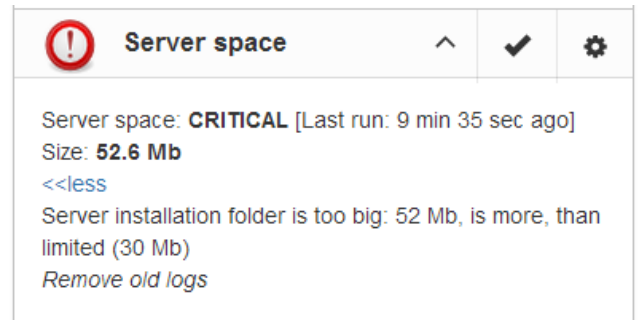
“Server space” jobs monitors size, occupied by Firebird installation.

It’s enabled by default.

There are several threats prevented by this job: misadministration issues when database volumes or external tables are being created in %Firebird%\Bin folder, very big **firebird.log** which can exhaust all places at drive with Firebird installation, and some other problems.

Also this job monitors and analyzes information, gathered by all space-related jobs (including database-level jobs).

At the picture below you can see quick representation of space analysis for all drives where Firebird, databases and backups are stored.



Using configuration dialog you can enable/disable this job, set period of checking and thresholds for server folder size.

By default we use 30Mb is a standard setting for Firebird installation. If the size of your Firebird is larger, please consider cleanup of old logs and other unwanted artifacts.

#### 4.9. Server: Send logs

“Send logs” is an auxiliary job which sends logs from server-level jobs by email with desired frequency.

This job is disabled by default.

If monitored database is situated at remote location it's useful to schedule logs sending by email. Using configuration dialog you can schedule logs sending

with CRON expression (for more details see [Appendix: CRON Expressions](#)), specify from what email it will be sent and where. Setting from email alerts settings will be used (for details see [3.5. Email alerts in FBDataGuard](#)).

Also you can specify logs to be sent by specifying appropriate Jobs IDs.

#### 4.10. Database: General configuration

FBDataGuard can monitor several databases at the single server.

For each database the separate widget is created.

At the top widget database status is shown, database nickname (it's specified during database adding and can be changed).

Also database widget shows the full path to the database, its size and status of backups.

#### Server installation space monitoring

Enabled

Check period, minutes:

10

Max occupied, %:

10

Max occupied, bytes:

31457280

#### Send database logs

Enabled

Schedule:

0 0 23 ? \* MON-FRI

E-mail from:

dataguard@here.com

E-mail to:

whoami@whereami.com

Jobs ids:

check-server-log, check-server-space, check-server-running, check-temp-files



production1: OK E:\DBW100\_FB252X64.FDB

Size: 9.0 Gb

Backups: OK (1/5)





Using configuration dialog you can set database nickname, path to database and output directory.

FBDataGuard checks validity of path to database and it does not allow entering wrong path.

In the title of dialog you can see GUID of this monitored database; it's used for unique identification of database and related logs.

Database monitoring configuration: a5081ce5-78c6-471f-8cbe-0a0de90f4ed2

Database name:	production1
DB alias:	
Path to database:	E:\DBW100_FB252X64.FDB
Output directory:	\$(server.default-directory)\\$(db.id)

#### 4.11. Database: Transactions

“Database: Transactions” job is intended to log transactions activity.

Later these logs can be analyzed to get helpful insight regarding database performance and application quality (see more information here <http://ib-aid.com/en/articles/ibanalyst-what-you-can-see-at-summary-view/>).

Also this job monitors for the implementation limit in Firebird: maximum transactions number should be less than  $2^{32}-1$ . Near this number database should be backup and restored. It will throw an alert if transaction number will be close to the restrictions.

#### Transactions monitoring

<input checked="" type="checkbox"/> Enabled	
Check period, minutes:	1
Max transactions:	192000000
Unusual transaction gap:	25000
Big transaction gap:	150000

#### 4.12. Database: Index statistics

“Database: Index statistics” is very important job which is related not only with indices and their performance, but also check for corruptions.

“Database: index statistics” allows to run re-computing of index selectivity value. During this procedure Firebird quickly walks through leaf pages of indices and renews statistics about selectivity. By visiting these pages Firebird also verifies their integrity and if index is corrupted, the warning will be thrown.

Also this job verifies that all indices are active in database. Inactive or non-activated indices usually indicate corruption and lead to performance degradation.

By default this job is disabled, but we recommend to enable it after careful selecting of indices to .

There are three modes in this job: AUTO, ALL, SELECTED. ALL is the mode where all indices will be checked. AUTO is the default mode. It is very similar to ALL, but it also checks the size of database and do not touch indices if database is bigger than 3.6Gb. SELECTED is the recommended mode. It allows choosing of indices which should be recomputed or those which should be avoided. To include

Update index stats configuration

Enabled

Schedule:  
0 0 22 ? \* MON-FRI

Update mode  
AUTO

Included indexes names

Excluded indexes names:

DB size to switch, bytes:  
4000000000

Check index activity:

Indices into the list of recomputed, you need to specify indices names (divided by comma), and to exclude – perform the same.

As you can see at configuration dialog screenshot, there are fields to enable/disable job, to set update mode, and to include or exclude indices. “DB size to switch, bytes” is to set limit where AUTO mode is working. “Check index activity” should be always on, until you are not performing special manipulations with inactive indices.

4.13. Database: Active users

“Database: Active users” checks for the number of active users and calculates minimum, maximum and average users count. It also stores users’ activity in the time logs, so it’s possible to see what was the situation with connections at particular period of time.

As you can see at job’s widget, “Database: Active users” checks shows minimum, average and maximum number for the period of last 24 hours.

Active users monitoring

Enabled

Check period, minutes:  
1

Minimum users:  
1

Maximum users:  
100

✔ Active Users
^
⚙

Active Users: **OK** [Last run: 58 sec ago]  
min/avg/max: 1/0.27/2  
last: 1

There are thresholds for minimum and maximum number of users. If your database should always have at list one (or N) connections (for example, it can be some automatic equipment), you can set the minimum user count you expect, and get warning when users count will be less than specified.

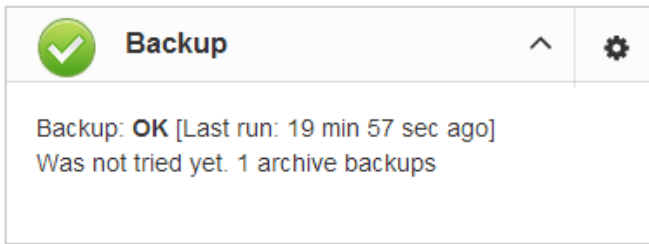
The same approach is for threshold for maximum number of connected users: for example, to catch peak connections times.

#### 4.14. Database: Backup

“Database: Backup” is one of the key jobs to guarantee the safety of data stored in the protected database. During development of FBDataGuard we had in mind certain recovery scenario, and this scenario implies that the key goal of database protection is to minimize losses of data. If we have healthy backup, recovery can be concentrated only the most recent data (just entered into the database), and it greatly decrease the time of overall outage.

As you will see below, “Database: Backup” is not just a wrapper for standard gbak functionality and scheduler, this is a smart job which has a lot built-in rules to prevent any problems with backups and provide suitable interface to manage backup.

**“Database: Backup” is enabled by default, but we strongly recommend reviewing of its setting immediately after FBDataGuard setup.**



Initially “Database: Backup” job is shown as Ok, though backup was not tried. In this case OK means that backup at least scheduled. Also this job recognize files according the name pattern (see below information regarding

configuration), and shows the totals number of backups.

After the backup will be done, the widget information will be changed: creation time of last succesfull backup will be shown, and also the time took to actually perform the backup (only 11 second at the screenshot with example).



“Database: Backup” checks the free space at the drive with backup destination, and if it detects that there is not enough free disk space, CRITICAL alert will be sent, and current backup will be canceled (if necessary).

**Be careful - by default backup time is set to 23-00 Monday-Friday.**

**By default database backups will be stored into the folder you specified during installation step!**

**It is very important to carefully review default database backups setting and adjust them according the local configuration!**

## Backups configuration

Enabled: <input checked="" type="checkbox"/>	Schedule: 0 25 19 ? * MON-FRI
Max duration, minutes: 120	Backups into: E:\temp
Backups archive depth: 5	Backup name pattern: backup_{0,date,yyyyMMdd_HH-mm}
Backup extension: .fbk	Compress backups: <input checked="" type="checkbox"/>
...with extension: .zbk	Check restore: <input checked="" type="checkbox"/>
Restore to: \${backup-directory}/restore.fdb.tmp	Remove restored: <input type="checkbox"/>
Copy backup: <input type="checkbox"/>	Copy backup to: /mnt/backups
Execute shell command: <input type="checkbox"/>	Shell command: 
Send 'ok' report: <input type="checkbox"/>	

At configuration dialog in the “Schedule” field you can set the time when backup should be run. Scheduler uses CRON expression and this is a right place to apply all the power of CRON (see [Appendix: CRON Expressions](#)).

“**Max duration**” time is a timeframe for backup timeout. Sometimes backup process can hang up, so exceeding of maximum time will immediately notify administrator of some problems. Also, if backup took too much time, it can mean problems with garbage collection or abnormal growth of database itself.

“**Backups into**” specifies the folder to store backups. This folder should be at the same computer where database is. By default it is situated inside database default directory. Usually it’s a good idea to set the explicit path to the folders with backups.

“**Backups archive depth**” specifies how many previous backups should be stored. FBDataGuard stores backups in revolver order: when the archive depth will be reached (i.e., 5 backups will be created), FBDataGuard will delete the oldest backup and create the new backup. In combination with CRON expressions it gives a powerful ability to create necessary history of backups.

“**Backup name pattern**” specifies how backup files will be named. Also this name pattern allows FBDataGuard to recognize old backups with the same name pattern.

“**Backup extension**” is fbk by default.

“**Compress backups**” specifies should FBDataGuard archive backups after regular Firebird backup. By default this option is on, but you need to know that FBDataGuard will zip backups’ files which are less than **12Gb** in size. After that backup compression will be automatically switched off. We recommend to turn this feature on for small databases or for database which work at non-dedicated servers (i.e., bundled with desktop applications, for example).

“**...with extension**” specifies extensions for compressed backup files.

“**Check restore**” is very important parameter. If it is on (by default), FBDataGuard will perform test restore of fresh backup, in order to test its validity. It guarantees the quality of created backups and notify administrator in case of any problems with test restore.

“**Restore to**” specifies the folder where to perform test restore. By default it’s inside database output folder. It’s a good idea to set the explicit path for test restore.

“**Remove restored**” specifies should FBDataGuard delete restored database. *By default it is OFF*, so you might want to turn it ON, but you need carefully consider – do you really need to keep the copy of test restored database. With each test restore this copy will be overwritten.

“**Copy backup**” switch and “**Copy backup to**” path. If you have network location or plugged USB drive to store database where you want to store copy of backup (in addition to usual backups), FBDataGuard can copy the latest backup there: just turn on “Copy backup” switch and specify “**Copy backup to**” path.

“**Execute shell command**” switch and “**Shell command**” path. It is possible to specify custom script or executable after the general backup procedure will be complete. Shell command gets as the path to the fresh database backup as a parameter.

“**Send “Ok” report**” – by default it is off, but it’s strongly recommended to turn it ON and start to receive notifications about correct backups. This feature will use email settings from alerts system (see [3.5. Email alerts in FBDataGuard](#)).

#### 4.15. Database: Store metadata

“Database: Store metadata” is one of the key jobs, it ensures database protection at low level. First of all, this job stores raw metadata in special repository, so in case of heavy corruption of database we can use repository

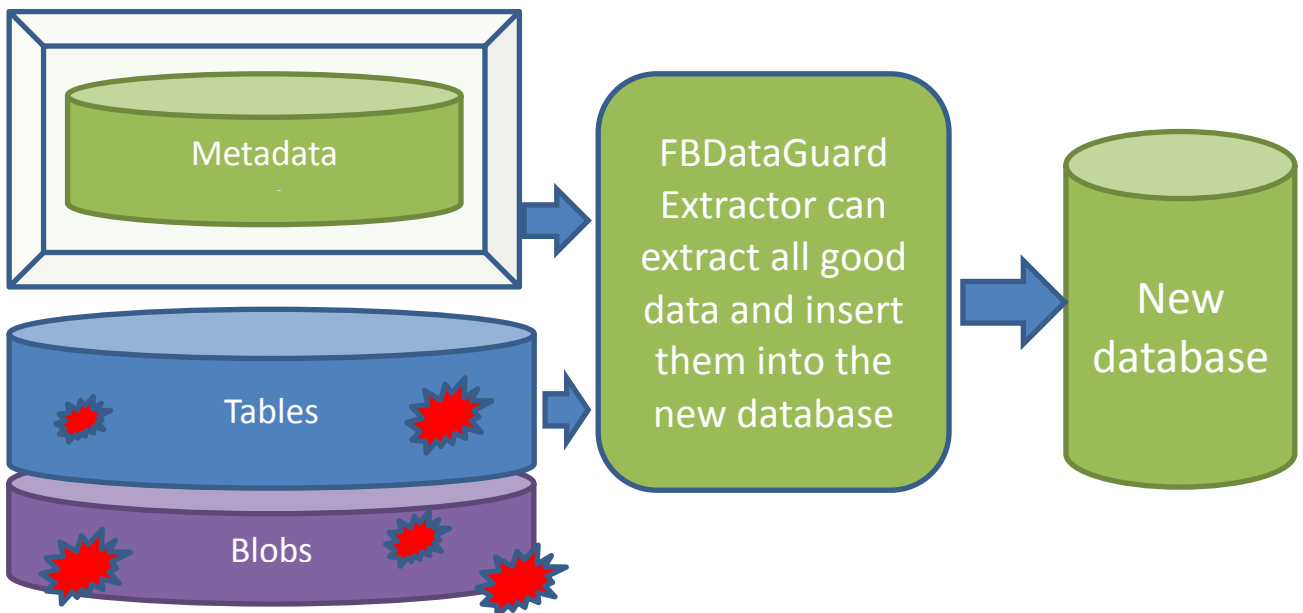


Figure 10 FBDataGuard Extractor can extract data from heavily corrupted databases

The second purpose of this job is constantly check tables for consistency. Every 20 minutes it pings all tables in the database and ensures there are no errors at metadata level.

The third purpose is to warn administrator about too many formats for each tables.

There is an implementation limit of 256 formats per table, but even several formats can greatly increase chance of hard corruption and can slow down the performance. It is recommended do not change tables structure at production database and keep only one format per each table. If it's not possible, administrator should try to perform backup/restore more often to transform all formats into the single one.

Database metadata backup configuration
✕

---

Enabled

Check period, minutes:

Store metadata in:

Date format for folder name:

Folder name prefix:

Max formats:

#### 4.16. Database: Validate DB

Validation of Firebird database requires exclusive access: i.e., no users should be connected during validation. “Database: Validate DB” job shuts down the database and performs validation of database, and then turns it on.

By default this job is OFF. Please consider carefully, is it possible to provide exclusive access for database. Validation can also take significant time. Using configuration dialog you can enable/disable this job, set time to run, set the shutdown timeout (time to wait before launch validation), and also

#### Update validation configuration

Update validation configuration
✕

---

Enabled

Schedule:

Shutdown timeout, sec:

Shutdown mode:

shutdown mode (FORCE, ATTACH, TRANSACTIONAL). If you have no deep knowledge n what you are doing, it's better to keep default parameters.

“Database: Validate DB” will send alert and put database to critical mode if there will be any errors. Also it is possible that these errors will be written into the firebird.log, so [4.6. Server: Server log job](#) will act accordingly.

#### 4.17. Database: Delta-files monitoring

If you are using nbackup, this job is critically important. It watches for delta-files lifetime and size, and warns if something goes wrong. Forgotten delta-files are the often reason of corruptions and significant losses of data.

This job finds all delta files associated with database and check their age and size. If one of these parameters exceeds thresholds “Maximum delta size” or “Maximum delta age”, administrator will receive the alert and database status will be set to CRITICAL.

*Note: If delta file from protected database was corrupted, it is possible to extract data from it using gathered metadata.*

#### Delta file monitoring

Enabled

Check period, minutes:

5

Maximum delta size, bytes:

52428800

Maximum delta age, minutes:

360

#### 4.18. Database: Disk space

This job watches for all objects related with database: database files (including volumes of multi-volume database), delta-files, backup files and so on.

“Database: Disk space” job analyze the growth of database and estimate will there be enough free space for the next operation like backup (including test restore) on the specific hard drive.

It generates several types of alerts. Problems with disk space are in the top list of corruption reasons, so please pay attention to the alerts from this job.

This job also contributes data to the server space analysis graph (4.8. [Server: Server space](#)).

By default this job is enabled.

Using configuration dialog you can specify check period and thresholds for free space. The first reached threshold will be alerted. To set threshold only in % of disk space, you need to set explicit space in bytes to “0”.

#### Space monitoring

Enabled:

Check period, minutes:

10

Free space minimum, %:

20

Free space minimum, bytes:

0

#### 4.19. Database: Database statistics

This job is very useful to capture performance problems and perform overall check of database at low-level without making backup.

We recommend running this job everyday and storing a history of statistics report. Using IBAAnalyst (<http://www.ibanalyst.com>) it is possible to discover deep problems with database performance. As a useful side effect, statistics visits all database pages for tables and indices, and ensures that all of them are correct.

#### 4.20. Database: Send logs

“Database: Send logs” is an auxiliary job which sends logs from database-level jobs by email with desired frequency.

This job is disabled by default.

If monitored database is situated at remote location it’s useful to schedule logs sending by email. Using configuration dialog you can schedule logs sending with CRON expression

(for more details see [Appendix: CRON Expressions](#)), specify from what email it will be sent and where. Setting from email alerts settings will be used (for details see [3.5. Email alerts in FBDataGuard](#)). Also you can specify logs from what jobs you need to send by specifying Jobs IDs.

## 5. FBDataGuard tips&tricks

FBDataGuard allows changing its setting not only through web-console, but also using direct modification of configuration files. This can be useful when you need to install FBDataGuard in silent mode (no interaction with user), to bundle it with third-party software, or to perform some fine configuration adjustments.

### 5.1. Path to FBDataGuard configuration

At start FBDataGuard looks for file **DataGuardJavaSvc.ini** – it should be located near executable DataguardJavaSvc.exe. In the end of this file you can find values

**param04 =-config-directory=G:\temp2\FBDataGuard\config**

**param05 =-default-output-directory=G:\temp2\FBDataGuard\output**

These values specify the paths to FBDataGuard configuration and output folder.

#### Database statistics configuration

Enabled

Schedule:

Store stats in:

Stats archive depth:

Stats file name pattern:

#### Send database logs

Enabled

Schedule:

E-mail from:

E-mail to:

Jobs ids:



If you setup FBDataGuard using Windows installer, these values are chosen during installation (see [2.3. Choosing folders to store configuration, backups and statistics](#))

If you wish to bundle FBDataGuard with third-party software, and build configuration manually, you need to write adequate paths in this ini.

## 5.2. Adjusting web-console port

One of the most frequently asked questions is how to adjust port for web-console application (by default it is 8082), It can be done by changing port setting in file

*%config%\agent\agent.properties*

**server.port = 8082 #change it**

*%config%* - folder to store configuration information, it is specified in [5.1. Path to FBDataGuard configuration](#)

## Appendix: CRON Expressions

All jobs in FBDataGuard have time settings in CRON format. CRON is very easy and powerful format to schedule execution times.

### CRON Format

A CRON expression is a string comprised of 6 or 7 fields separated by white space. Fields can contain any of the allowed values, along with various combinations of the allowed special characters for that field. The fields are as follows:

Field Name	Mandatory	Allowed Values	Allowed Special Characters
Seconds	YES	0-59	, - * /
Minutes	YES	0-59	, - * /
Hours	YES	0-23	, - * /
Day of month	YES	1-31	, - * ? / L W
Month	YES	1-12 or JAN-DEC	, - * /
Day of week	YES	1-7 or SUN-SAT	, - * ? / L #
Year	NO	empty, 1970-2099	, - * /


So cron expressions can be as simple as this: \* \* \* \* ? \*

or more complex, like this: 0 0/5 14,18,3-39,52 ? JAN,MAR,SEP MON-FRI 2002-2010


### Special characters

- \* ("*all values*") - used to select all values within a field. For example, "\*" in the minute field means "*every minute*".
- ? ("*no specific value*") - useful when you need to specify something in one of the two fields in which the character is allowed, but not the other. For example, if I want my trigger to fire on a particular day of the month (say, the 10th), but don't care what day of the week that happens to be, I would put "10" in the day-of-month field, and "?" in the day-of-week field. See the examples below for clarification.
- - - used to specify ranges. For example, "10-12" in the hour field means "*the hours 10, 11 and 12*".
- , - used to specify additional values. For example, "MON,WED,FRI" in the day-of-week field means "*the days Monday, Wednesday, and Friday*".
- / - used to specify increments. For example, "0/15" in the seconds field means "*the seconds 0, 15, 30, and 45*". And "5/15" in the seconds field means "*the seconds 5, 20, 35, and 50*". You can also specify '/' after the " **character - in this case** " is equivalent to having '0' before the '/'. '1/3' in the day-of-month field means "*fire every 3 days starting on the first day of the month*".

- **L** ("*last*") - has different meaning in each of the two fields in which it is allowed. For example, the value "L" in the day-of-month field means "*the last day of the month*" - day 31 for January, day 28 for February on non-leap years. If used in the day-of-week field by itself, it simply means "7" or "SAT". But if used in the day-of-week field after another value, it means "*the last xxx day of the month*" - for example "6L" means "*the last friday of the month*". When using the 'L' option, it is important not to specify lists, or ranges of values, as you'll get confusing results.
- **w** ("*weekday*") - used to specify the weekday (Monday-Friday) nearest the given day. As an example, if you were to specify "15W" as the value for the day-of-month field, the meaning is: "*the nearest weekday to the 15th of the month*". So if the 15th is a Saturday, the trigger will fire on Friday the 14th. If the 15th is a Sunday, the trigger will fire on Monday the 16th. If the 15th is a Tuesday, then it will fire on Tuesday the 15th. However if you specify "1W" as the value for day-of-month, and the 1st is a Saturday, the trigger will fire on Monday the 3rd, as it will not 'jump' over the boundary of a month's days. The 'W' character can only be specified when the day-of-month is a single day, not a range or list of days.

 The 'L' and 'W' characters can also be combined in the day-of-month field to yield 'LW', which translates to "*last weekday of the month*".

- **#** - used to specify "the nth" XXX day of the month. For example, the value of "6#3" in the day-of-week field means "*the third Friday of the month*" (day 6 = Friday and "#3" = the 3rd one in the month). Other examples: "2#1" = the first Monday of the month and "4#5" = the fifth Wednesday of the month. Note that if you specify "#5" and there is not 5 of the given day-of-week in the month, then no firing will occur that month.

 The legal characters and the names of months and days of the week are not case sensitive. MON is the same as mon.

## CRON Examples

Here are some full examples:

Expression	Meaning
0 0 12 * * ?	Fire at 12pm (noon) every day
0 15 10 ? * *	Fire at 10:15am every day
0 15 10 * * ?	Fire at 10:15am every day
0 15 10 * * ? *	Fire at 10:15am every day
0 15 10 * * ? 2005	Fire at 10:15am every day during the year 2005
0 * 14 * * ?	Fire every minute starting at 2pm and ending at 2:59pm, every day
0 0/5 14 * * ?	Fire every 5 minutes starting at 2pm and ending at 2:55pm, every day
0 0/5 14,18 * * ?	Fire every 5 minutes starting at 2pm and ending at 2:55pm, AND fire every 5 minutes starting at 6pm and ending at 6:55pm, every day

0 0-5 14 * * ?	Fire every minute starting at 2pm and ending at 2:05pm, every day
0 10,44 14 ? 3 WED	Fire at 2:10pm and at 2:44pm every Wednesday in the month of March.
0 15 10 ? * MON- FRI	Fire at 10:15am every Monday, Tuesday, Wednesday, Thursday and Friday
0 15 10 15 * ?	Fire at 10:15am on the 15th day of every month
0 15 10 L * ?	Fire at 10:15am on the last day of every month
0 15 10 ? * 6L	Fire at 10:15am on the last Friday of every month
0 15 10 ? * 6L	Fire at 10:15am on the last Friday of every month
0 15 10 ? * 6L 2002-2005	Fire at 10:15am on every last Friday of every month during the years 2002, 2003, 2004 and 2005
0 15 10 ? * 6#3	Fire at 10:15am on the third Friday of every month
0 0 12 1/5 * ?	Fire at 12pm (noon) every 5 days every month, starting on the first day of the month.
0 11 11 11 11 ?	Fire every November 11th at 11:11am.



Pay attention to the effects of '?' and '\*' in the day-of-week and day-of-month fields!

## Notes

- Support for specifying both a day-of-week and a day-of-month value is not complete (you must currently use the '?' character in one of these fields).
- Be careful when setting fire times between mid-night and 1:00 AM - "daylight savings" can cause a skip or a repeat depending on whether the time moves back or jumps forward.

More information is here

<http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html>

## 6. Support contacts

We will answer all your questions regarding FBDataGuard. Please send enquiries to [support@ib-aid.com](mailto:support@ib-aid.com).