



## Firebird White Paper

# Minimizing database size: should you store data files in the file system?

Holger Klemt, February 2018

One of the most significant Firebird features is the effective management of BLOB information.



The 1958 movie “The BLOB” headlined “Indescribable... Indestructible! Nothing can stop it!” is not particularly helpful. Fortunately, it has nothing to do with BLOBs in database systems. The “Thing” from the movie starring Steve McQueen is a kind of killer amoeba from space that eats and digests anything that comes in its way.

A BLOB in a Firebird database can be better understood as a “Binary Large Object”. In the context of data management this includes everything that falls under the concept of a data file. Pretty well any binary data chain of any size can be stored on the file system for later usage. Simple text files, pictures, videos and much more are suitable for this kind of storage.

For documentation in ERP (Enterprise Resource Planning) and CRM (Customer Relationship Management) systems, more and more people are switching to not just printing the documents directly to a printer, but also storing them as PDF files in order to send them, for example, as an email attachment.

A lot of data can come together during regular day time operations. For a customer with approximately 40 employees, who uses our BRP software for order processing and production planning, at least 500 PDF documents per day are produced. These documents

can contain several individual pages including photos necessary for the production of their projects.

This customer has been deploying our BRP software in their production environment for 6 years, and over this timeframe the average document size has been determined to be 500 KB. Today approximately 600,000 documents with an average size of 0.5 MB are stored in the archive.



Many clients still think that there are advantages to store such data elements in the file system. Our experience with Firebird has shown us that the data storage in the database offers significant advantages.

1. Try to store 1 million files in a folder on a NTFS-Filesystem. You will quickly discover that tools like Windows Explorer are pretty useless, because the folder is barely able to open.
2. Try to prevent a file from being overwritten or deleted by a user, even though the user has write permission to the folder. All changes should be transactionally logged and undone, if needed. The recycle bin is rapidly overwhelmed due to the file size, and emptied quickly or is not even available on the network drive.
3. Try to store multiple versions of the same file under the same name simultaneously, keeping track of who did what and when? You will find that a number of problems arise, especially in the network.
4. Despite appropriate write permissions, try to prevent files from being moved to other folders. Your system will collapse faster than you think, and nobody knows who was responsible.
5. If a user accidentally starts an email attachment with one of the not so rare encryption trojans, you probably will not notice that this has already partially encrypted all important files on reachable file shares in the network and thus made them useless.
6. Prevent an employee from simply copying onto a local drive, and taking home all the documents he has network access to ... perhaps as a reminder of his former client relationships, which he would like to use at his new employer's ... difficult, as Edward Snowden and other have shown.
7. Try to search for a file in the file system. In the age of modern SSDs, the response time is not quite as catastrophic as it is without it, but if you also search through content, you should plan minutes or even hours, as the availability of functional indices is limited.

There are many other arguments against file system storage, all convincing us that storing files in the file system is wrong. Certainly there are possibilities in active directory or Linux file systems to limit the basic problems by restrictions. But protection against the above mentioned problems is hardly feasible.

In our IBExpert Firebird 3 Bootcamps, we will present a combination of the following technologies, and you will see that storing millions of PDFs or images in Firebird databases enables an extremely powerful integration of DMS (Data Management Technologies) in your application.

- How do I read and write BLOB data with calculated columns and updatable views?
- How do I save and read the BLOB data in a separate database, without changes in my application and avoiding a too large production database?
- How do I move the BLOB data from my BLOB database to a read-only history database on one or more different servers in an annual or monthly process?
- How do I process PDF files without any Adobe tools to preview the first or all pages?
- How do I extract the text from a PDF file without Adobe tools?
- How can I view PDF data in my Windows application without using Adobe tools?
- How do I integrate a powerful OCR engine to create text, images, or non-text image PDFs?
- How can I create a powerful full-text search engine based on these techniques?



# IBEXPERT GMBH WHITE PAPER



If your time does not allow you to visit our IBExpert Firebird 3 Bootcamp, we offer a 12-hour remote training and project support by phone and TeamViewer/pcvisit/GotoMeeting. The training takes place in 3 daily sessions of 4 hours each. The price for this remote course is \$ 2,480 USD.

## In conclusion...

Steve McQueen was offered \$ 2,500 or 10% of the profits. He took the \$ 2,500 because the film wasn't expected to make much. It ended up grossing over \$ 4 million. According to producer Jack H. Harris when being interviewed by film historian Tom Weaver, the film ultimately grossed \$ 40 million.

**... so you should not underestimate the value of BLOBs!**

### Contact:

IBExpert  
202 River Bend CT  
32779 Longwood FL.  
USA

Telephone: **407-462-7283** Monday to Friday 9am-6pm  
Email: [contact@ibexpert.com](mailto:contact@ibexpert.com)

