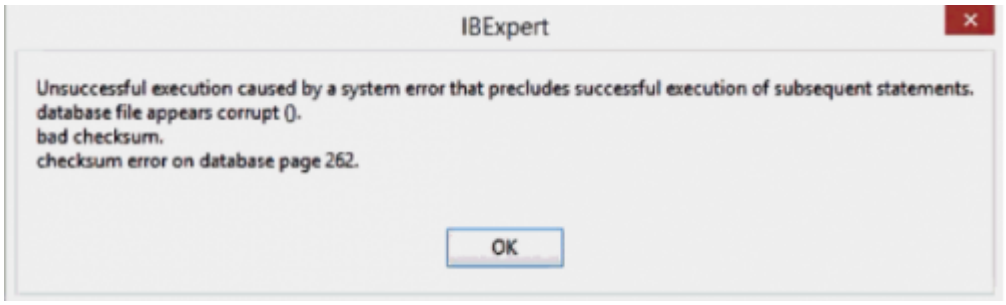
Database Inside

Database Inside can be found in the Tools menu in the full version of IBExpert. You can use the IBExpert *Database Inside* feature to analyze and repair databases. (This feature is unfortunately not included in the free IBExpert Personal Edition.)

Database Inside reads the database file directly, without a server. This allows extraction of data/metadata from corrupted databases even if it is impossible to do this using a normal connection to the database. The result depends on how heavily the database is corrupted. *Database Inside* processes files in read-only mode so files remain unchanged.

IBExpert also supports Firebird 3 databases as long as they are not encrypted.

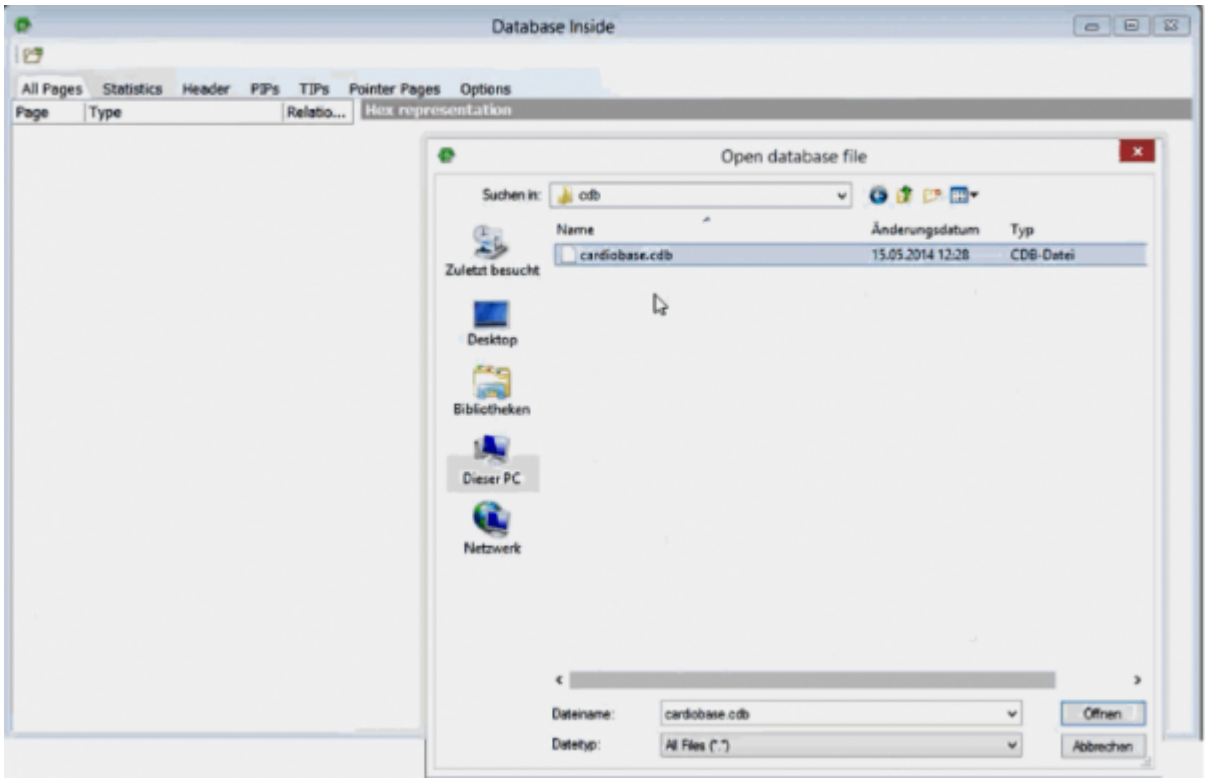
Here we have a database, which is corrupt. When attempting to connect to the database, an error message appears which reads: *Bad Checksum error on database page 262*:



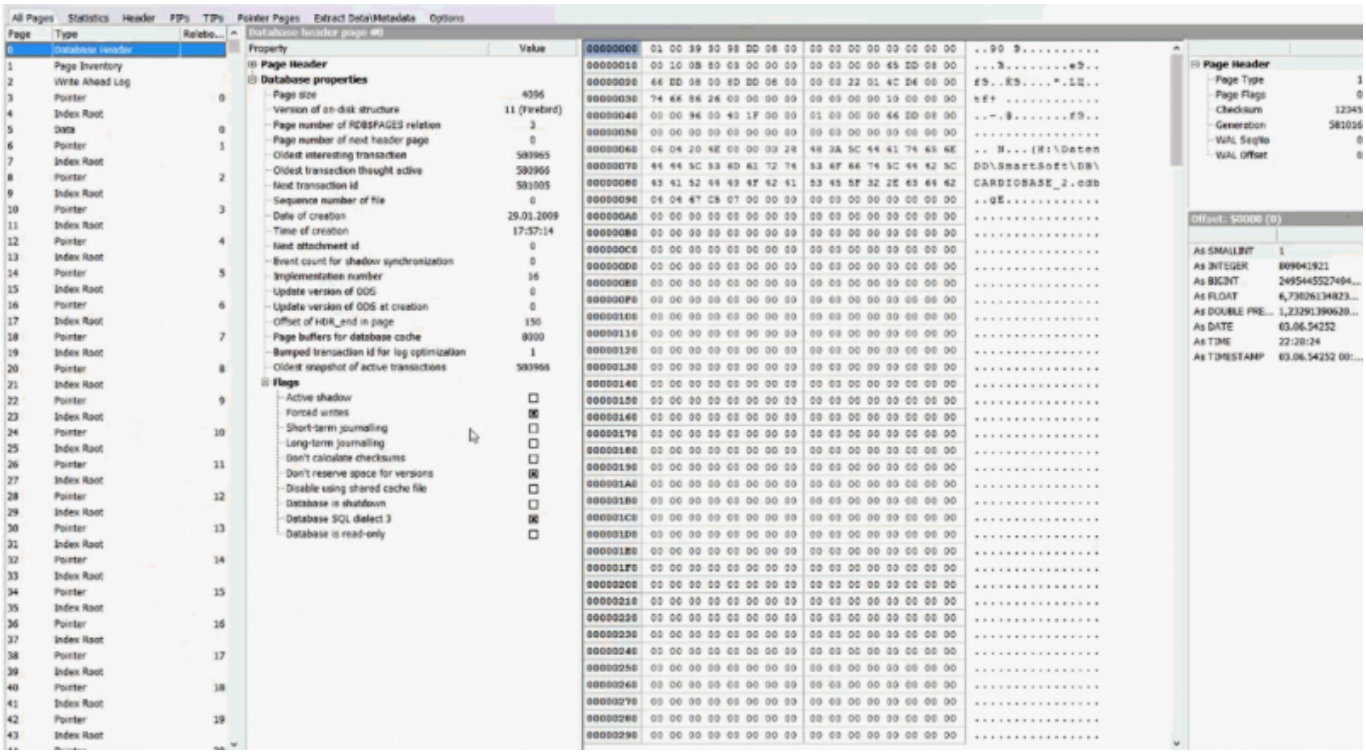
It is not even possible to perform a successful backup – even without the Garbage Collection or with or without any other options. In this example, one of the system tables is damaged; so it's a pretty serious problem. Even using gfix to validate the database is not successful.

So, we can either discard the database or use *Database Inside*.

When you start *Database Inside* you are asked to select and subsequently open a database file:



The selected database is opened and loaded accordingly:



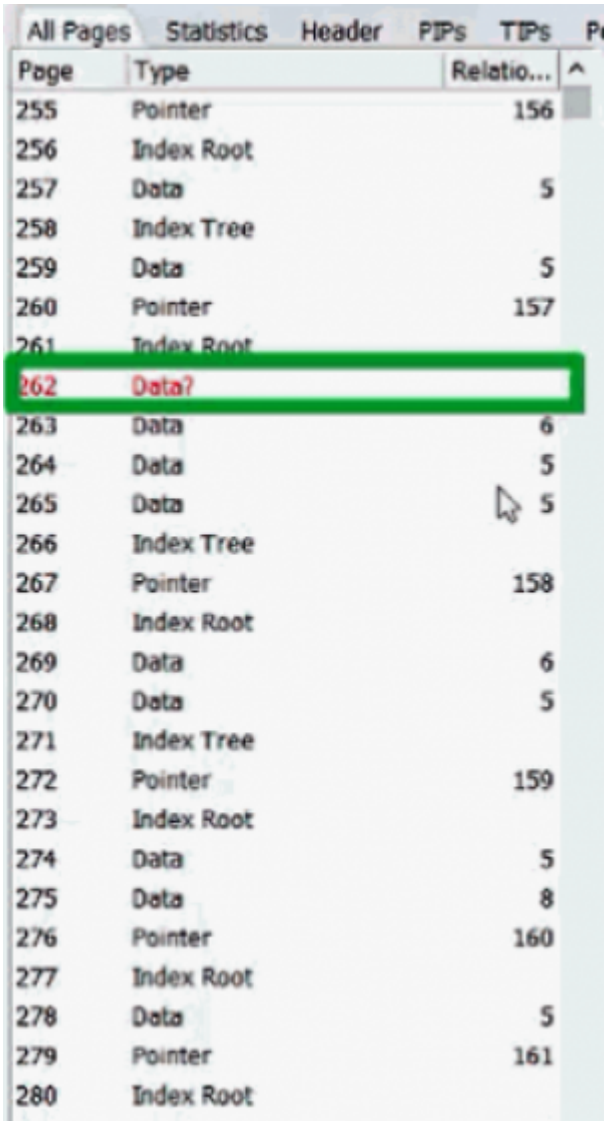
Important: there is no Firebird server or tool started or involved here in any way. IBExpert's internal structures read and load the database content. This allows you to view and repair databases that are otherwise irreparable.

back to top of page

All pages

The *All pages* page displays a list of database pages in natural order. Page types to be displayed here are customizable on the Options page. Please note that for large databases it is better to disable the display of the *index tree*, *index root* and *blob data* pages in order to optimize memory usage.

Suspicious pages (pages marked as allocated with unknown page type or/and wrong checksum) appear highlighted in **red**.



Here we can see that page 262 – just as Firebird has already told us – is corrupt.

When you click through the page types listed on the left, you will see that different information according to page type is displayed. For example, when clicking on the **Database header** page in the *All pages* list, you can view page header information, database properties and flags as hex code, simply by clicking on the item.

Double click on the page or press *Enter* to open the data page in a separate window.

Please note, that when a data page or pointer page is selected, there are three data views in the lower part of the screen: *Columns view*, *Raw data*, *packed* and *Unpacked data*.

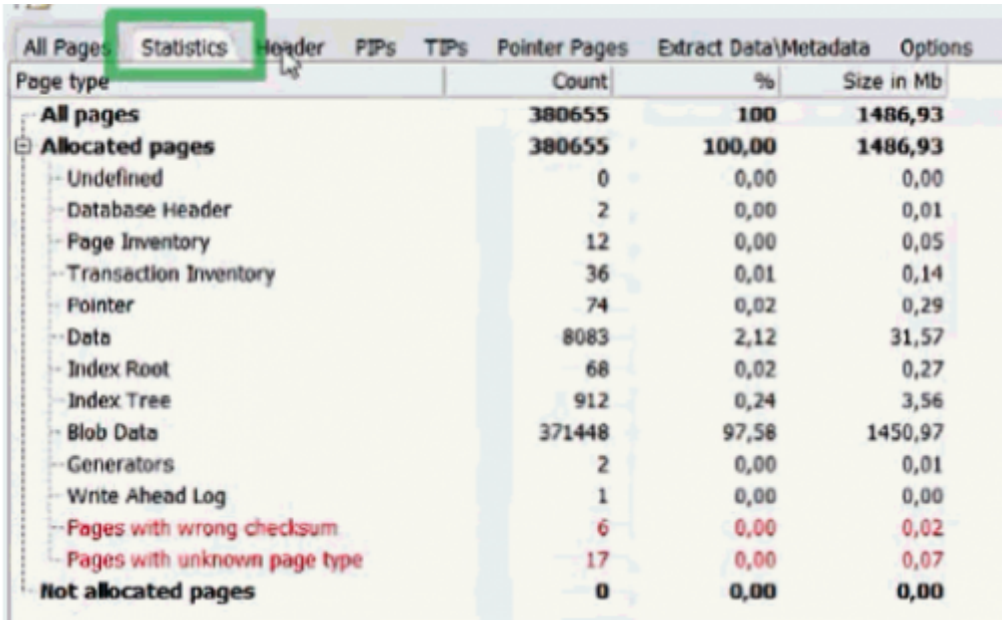
You can read more about data pages and pointer pages in the following articles:

- Pointer Page - type 0x04
- Data Page - type 0x05
- Pointer page (PTR)
- Data page (DPG)
- Structure of a data page

back to top of page

Statistics

The *Statistics* page contains some useful statistics related to distribution of pages within the database file. Suspicious data (number of allocated pages with undefined/unknown page type and/or wrong checksum) appears highlighted in **red**.



Page type	Count	%	Size in Mb
All pages	**380655**	**100**	**1486,93**
Allocated pages	**380655**	**100,00**	**1486,93**
Undefined	0	0,00	0,00
Database Header	2	0,00	0,01
Page Inventory	12	0,00	0,05
Transaction Inventory	36	0,01	0,14
Pointer	74	0,02	0,29
Data	8083	2,12	31,57
Index Root	68	0,02	0,27
Index Tree	912	0,24	3,56
Blob Data	371448	97,58	1450,97
Generators	2	0,00	0,01
Write Ahead Log	1	0,00	0,00
Pages with wrong checksum	6	0,00	0,02
Pages with unknown page type	17	0,00	0,07
Not allocated pages	**0**	**0,00**	**0,00**

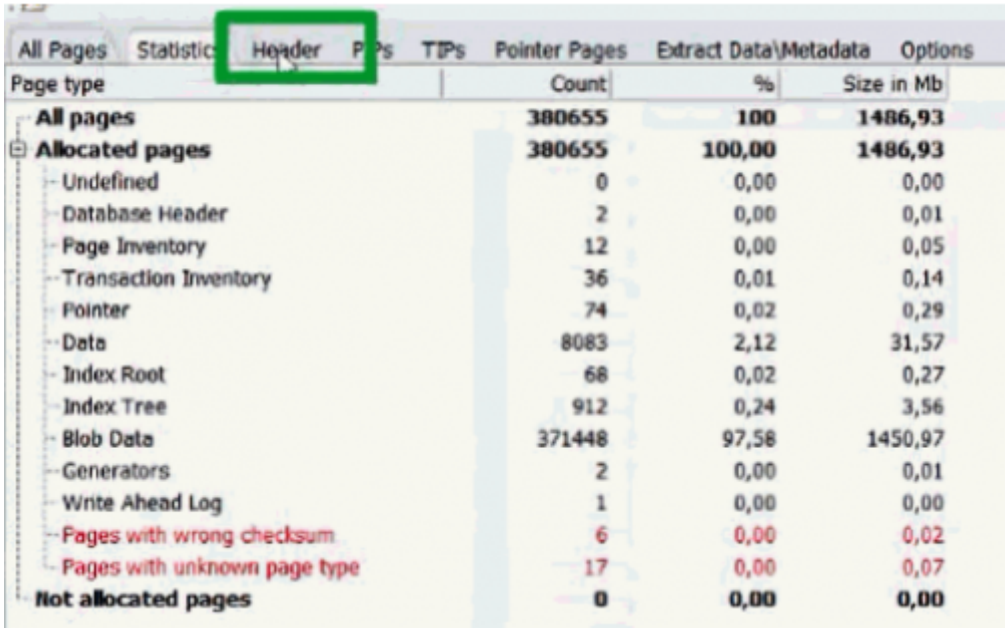
For further information regarding the individual page types, refer to the Database Technology article, Firebird for the database expert: page types and the IBExpert documentation chapter, Database Statistics.

Definition: WAL - Write Ahead Log

back to top of page

Header

The *Header* page contains the database header data.



Page type	Count	%	Size in Mb
All pages	**380655**	**100**	**1486,93**
Allocated pages	**380655**	**100,00**	**1486,93**
Undefined	0	0,00	0,00
Database Header	2	0,00	0,01
Page Inventory	12	0,00	0,05
Transaction Inventory	36	0,01	0,14
Pointer	74	0,02	0,29
Data	8083	2,12	31,57
Index Root	68	0,02	0,27
Index Tree	912	0,24	3,56
Blob Data	371448	97,58	1450,97
Generators	2	0,00	0,01
Write Ahead Log	1	0,00	0,00
Pages with wrong checksum	6	0,00	0,02
Pages with unknown page type	17	0,00	0,07
Not allocated pages	**0**	**0,00**	**0,00**

Again, when you click through the list on the left, you can see the exact hex code used by Firebird to define this property.

There are a number of interesting articles about the Header Page:

- Database Header Page - type 0x01
- Standard database page header
- Firebird for the database expert: Header page (HDR)
- Structure of a header page

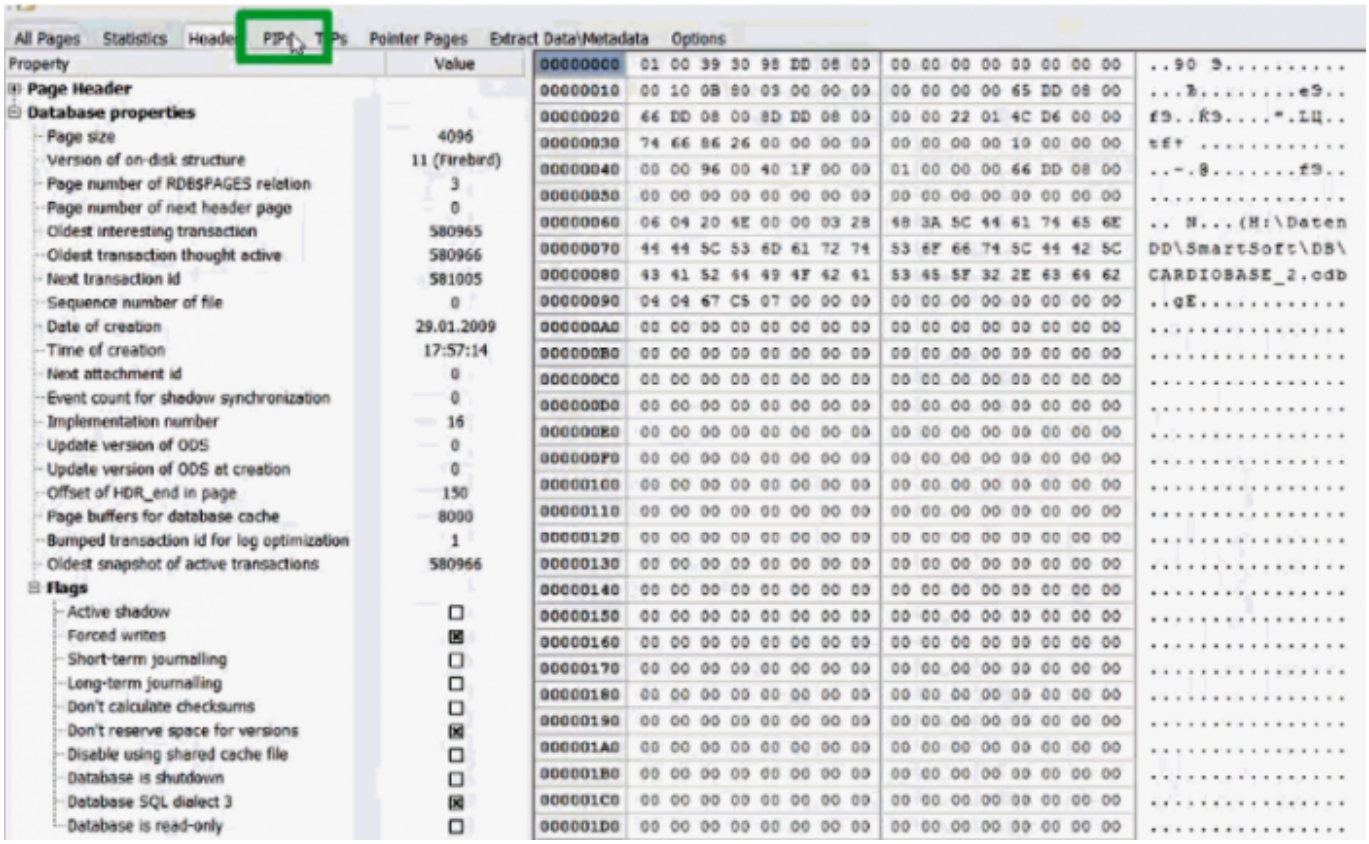
and the subject, *Database Properties* is dealt with in detail in the IBExpert chapter, Database Properties.

Definition: OIT - Oldest Interesting Transaction

back to top of page

PIPs

The *PIPs* (PIP: Page Inventory Pages) page contains list of all PIPs in the database file and information about the allocation of pages.



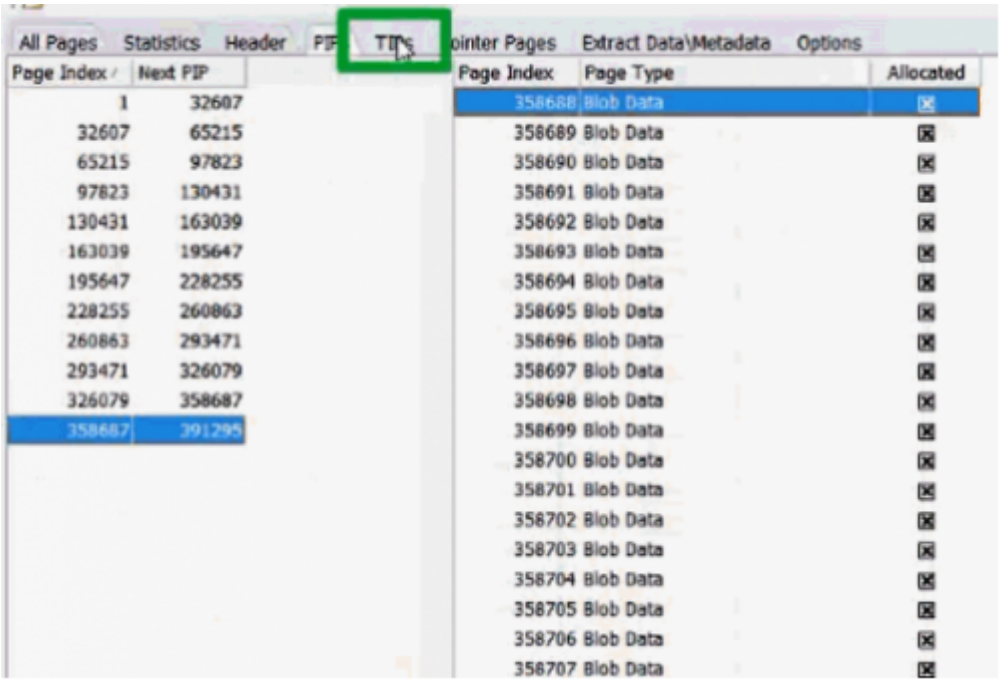
You can read more about PIPs in the following articles:

- Firebird for the Database Expert: Episode 2 - Page Types
- Firebird Internals: Page Inventory Page

back to top of page

TIPs

The *TIPs* (TIP: transaction inventory pages) page contains list of all TIPs in the database.



You can read more about TIPs in the following articles:

Firebird for the database expert: Transaction Inventory Page (TIP) Transaction Inventory Page - type 0x03

back to top of page

Pointer pages

The *Pointer pages* page contains list of all pointer pages for each relation in the database.

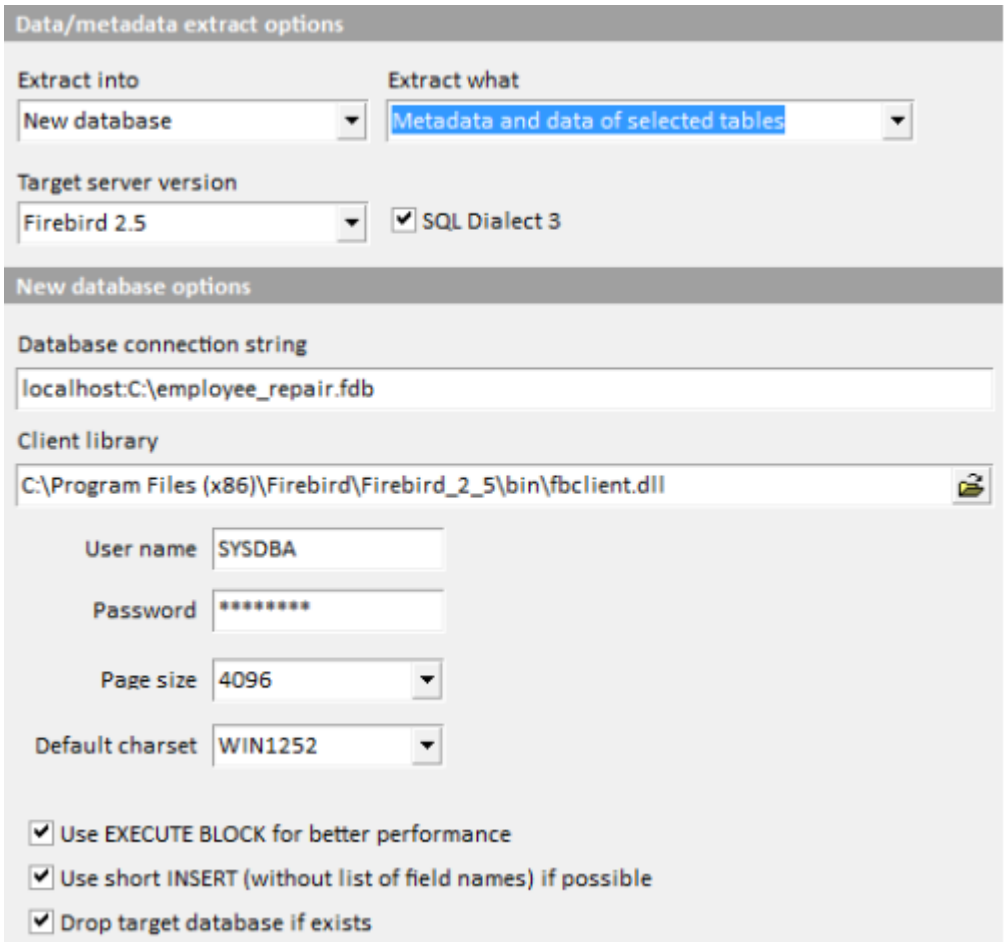
You can read more about PTRs in the following articles:

Firebird for the database expert:: Pointer page (PTR) Pointer Page - type 0x04

back to top of page

Extract data/metadata

The *Extract Data/Metadata* page allows you to extract data and/or metadata directly from the database file. Metadata/data can be extracted into a set of SQL script files or into a new database (a working server instance is necessary in this case). You can extract just metadata, just data or both. It is important to specify the correct Firebird server version and SQL dialect. And don't forget to specify a *Target directory* if you are exporting to script files.



In our example here we've uses the relevant Embedded Firebird client. We could of course use a normal Firebird client; this would then run as a client via a remote connection.

To speed up the whole thing we've checked the option, Use *EXECUTE BLOCK for better performance* and – just in case the database already exists, we've checked the option, *Drop target database if exists*.

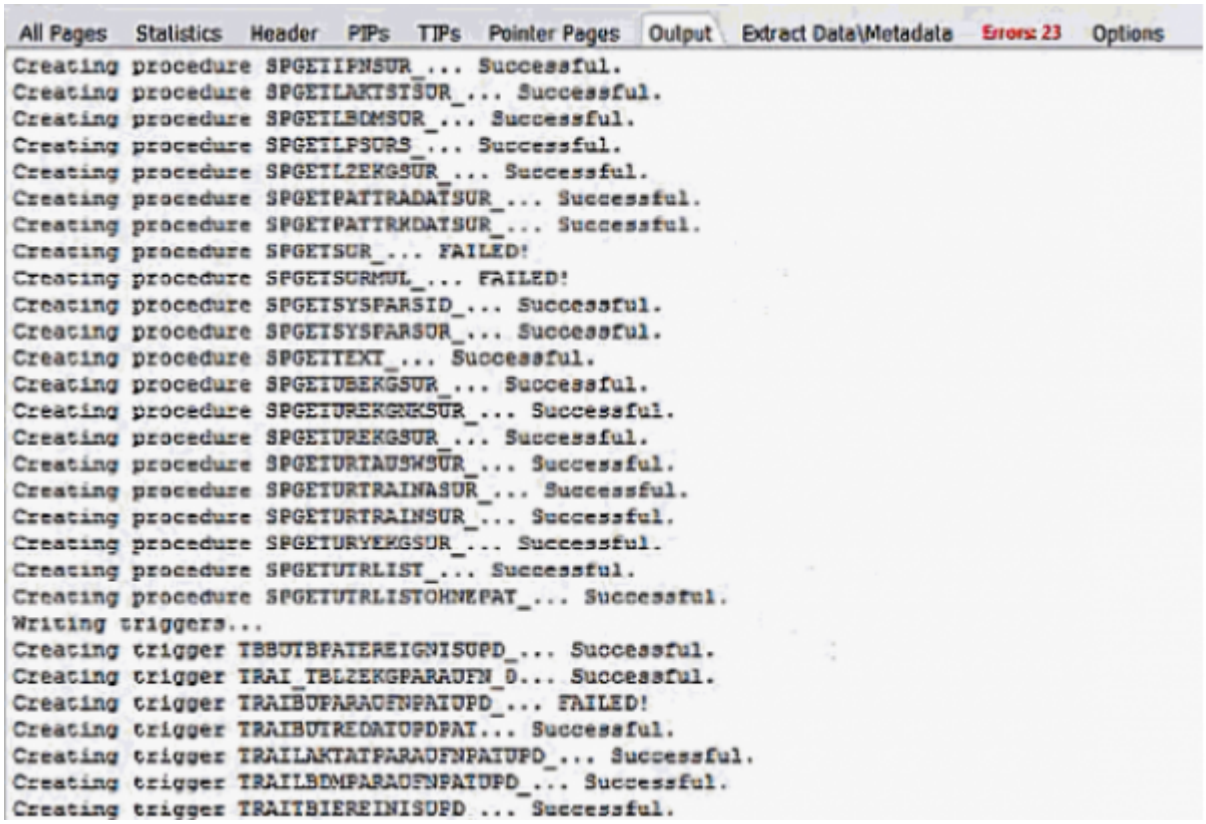
The option *Use short INSERT if possible option* generates a short version of INSERT - without the list of columns - as opposed to the long INSERT. This allows you to put more statements in a block and improve performance.

You can select/unselect all data tables simultaneously from the context menu of the data tables list.

Now simply click *Extract data/metadata* in the bottom right-hand corner to save as much of your database as is possible!

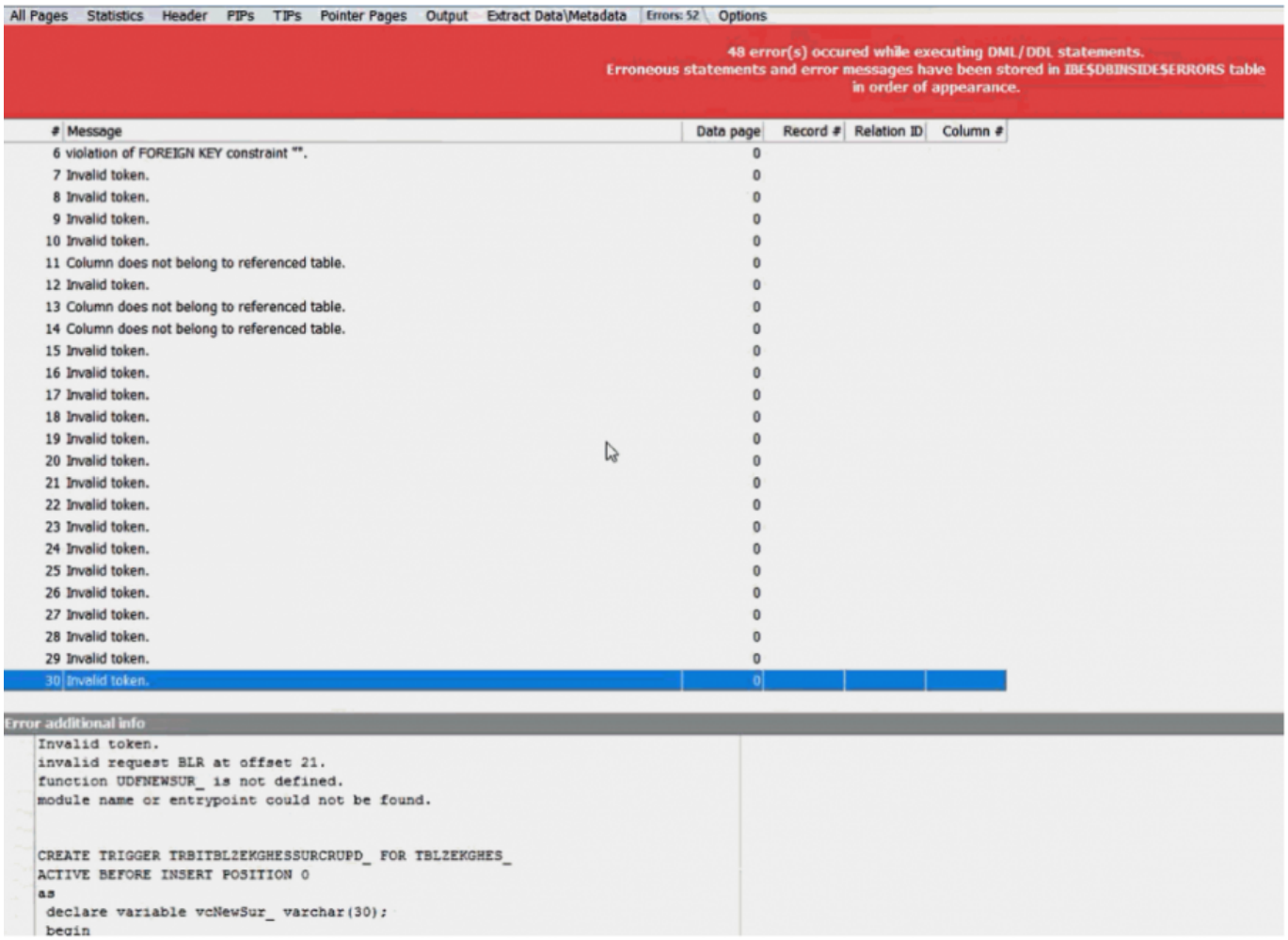
IBExpert now opens the Firebird database in binary form. We are not touching any Firebird instance at all. This way we have the possibility to bypass all of Firebird's security mechanisms and internal structures. IBExpert protocols the export process, detailing which data objects for example, cannot be recreated without errors.

This is vital, in order to determine which parts of the database still function correctly and where any problems lie.



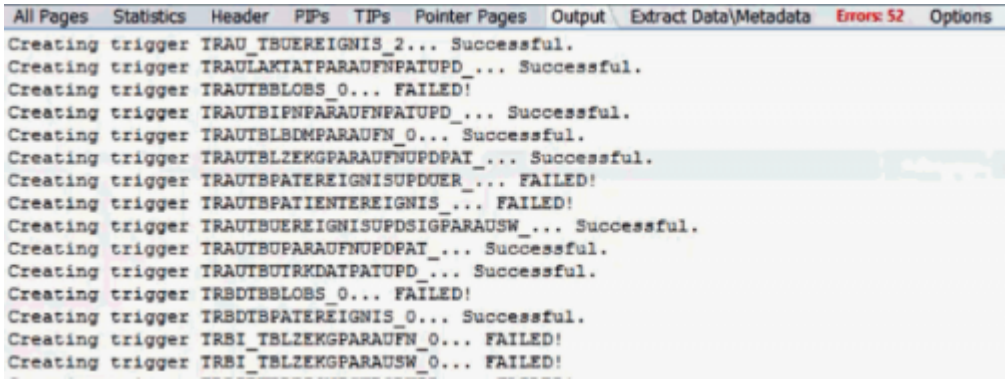
The metadata objects are created, and this metadata is displayed by IBExpert as a summarized evaluation at the end. IBExpert shows you where it was not possible to execute certain items, for example which metadata objects could not be processed correctly.

All commands executed here are included in the IBE$DBINSIDE$ERRORS table. This can be opened separately from IBExpert's Database Explorer, once you have registered the repaired database for use in IBExpert.



You can then normally trace and carry out any necessary repair work, for example going to the indicated location in certain objects, or trace to, for example, certain UDFs in the UDF library that are referred to here in our example, which are not usable.

You can see quite clearly the list of operations that were successful along with those that were unsuccessful:



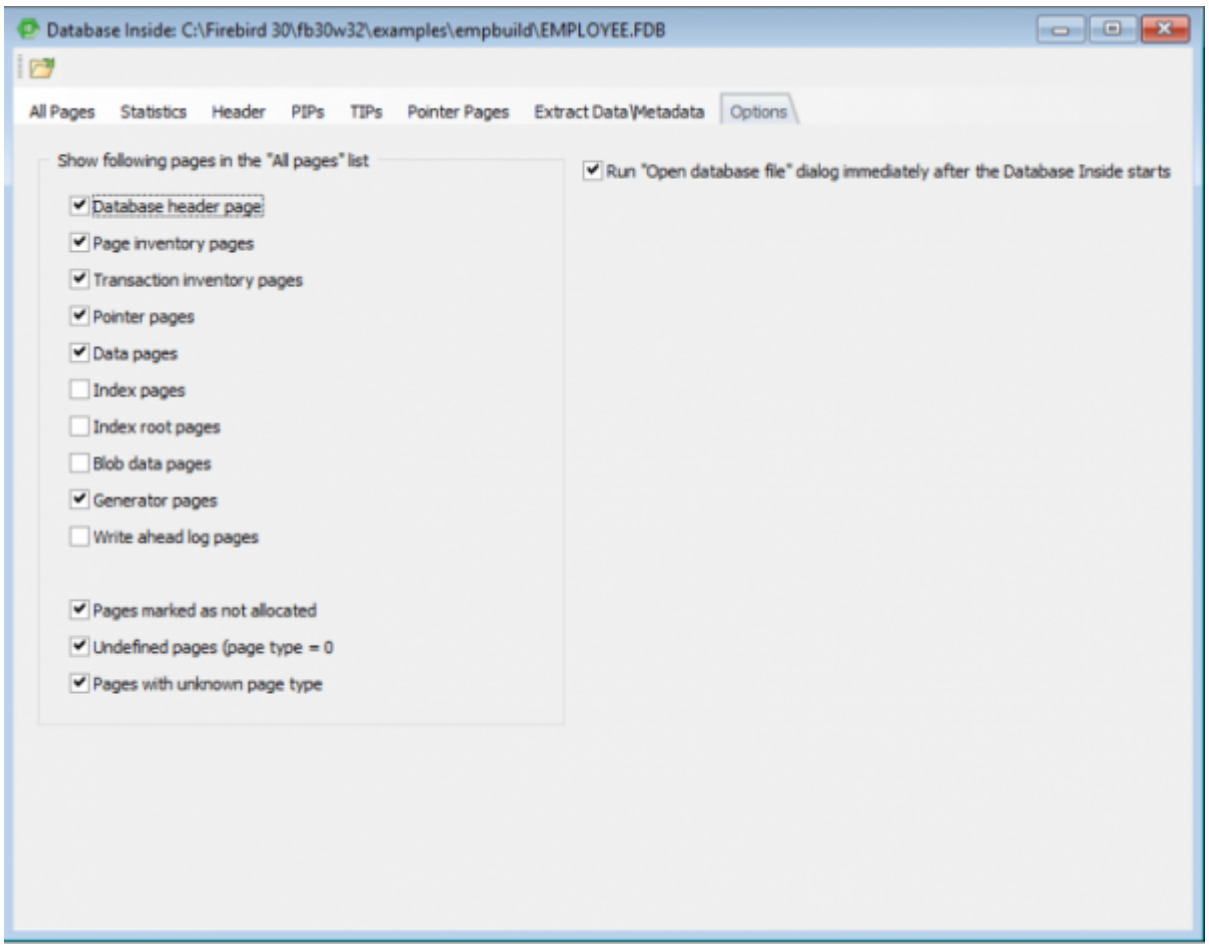
You can then register the repaired database for use in IBExpert.

Learn more about extracting metadata in the IBExpert documentation chapter, Extract metadata.

back to top of page

Options

On the *Options* page you can select the types of pages to be listed in All pages page.



Please note that for large databases it is better to disable the display of the *index tree*, *index root* and *blob data* pages in order to optimize memory usage.

back to top of page

Current limitations

- only single-file databases are supported at the moment.
- InterBase® databases with object names' length > 31 chars are not supported yet.
- the *Database Inside* feature has been tested with Firebird 1.5-2.5 databases created with Firebird for Windows.

Further reading

- Database corruption
- Database Validation

- Tracking down crashes on Linux
- Tracking down crashes on Win32 systems
- How to analyze and repair a corrupted database
- Preventing data loss
- Database repair