

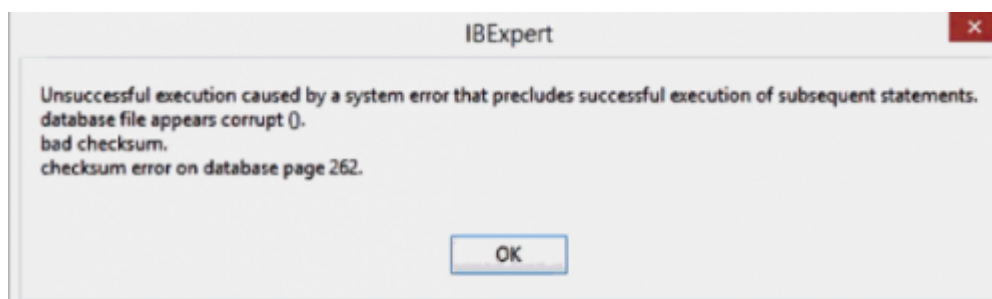
Database Inside

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.

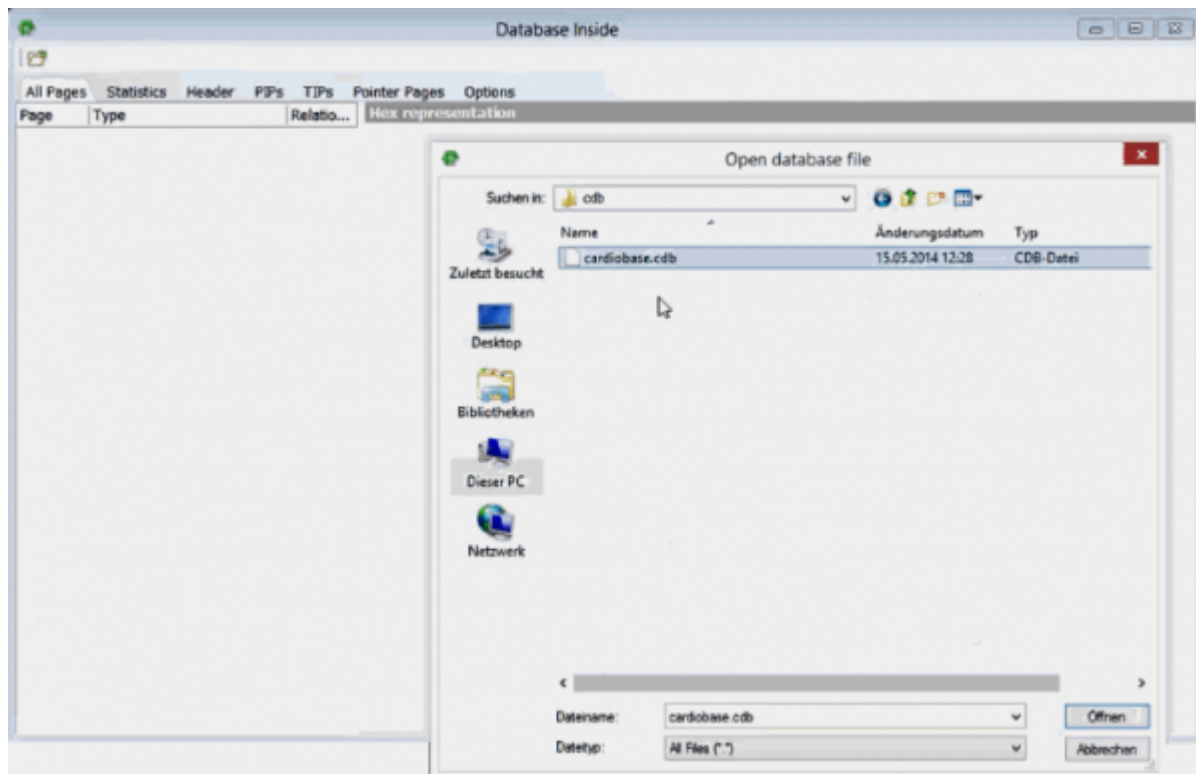
Here we have a database, which is corrupt. When attempting to connect to the database, an error message appears which reads: *Bad Checksum error on database page 262*:



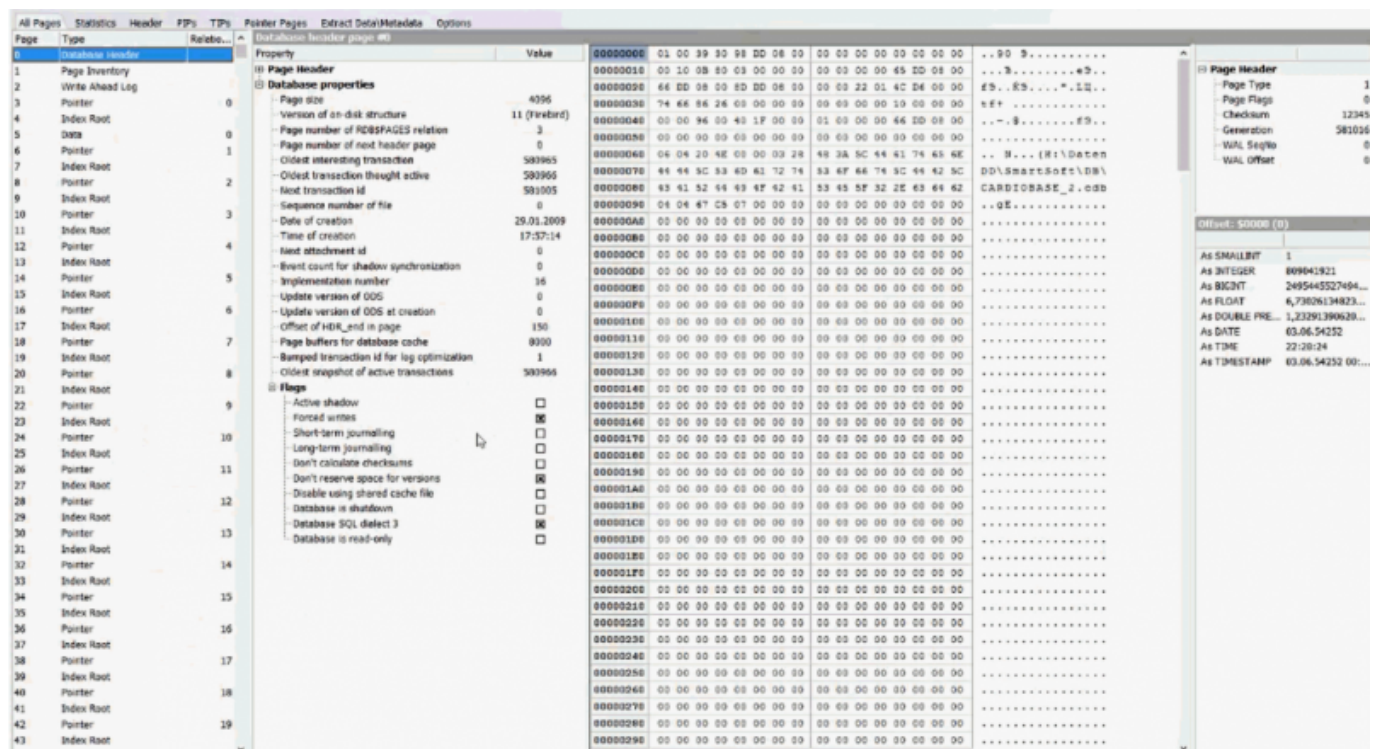
It is not even possible to perform a successful backup – even without the [Garbage Collection](#) or with or without any other options. In this example, one of the system tables is damaged; so it's a pretty serious problem. Even using [gfix](#) to validate the database is not successful.

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**.

Page	Type	Relatio...	^
255	Pointer	156	
256	Index Root		
257	Data	5	
258	Index Tree		
259	Data	5	
260	Pointer	157	
261	Index Root		
262	Data?		
263	Data	6	
264	Data	5	
265	Data	5	
266	Index Tree		
267	Pointer	158	
268	Index Root		
269	Data	6	
270	Data	5	
271	Index Tree		
272	Pointer	159	
273	Index Root		
274	Data	5	
275	Data	8	
276	Pointer	160	
277	Index Root		
278	Data	5	
279	Pointer	161	
280	Index Root		

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*.

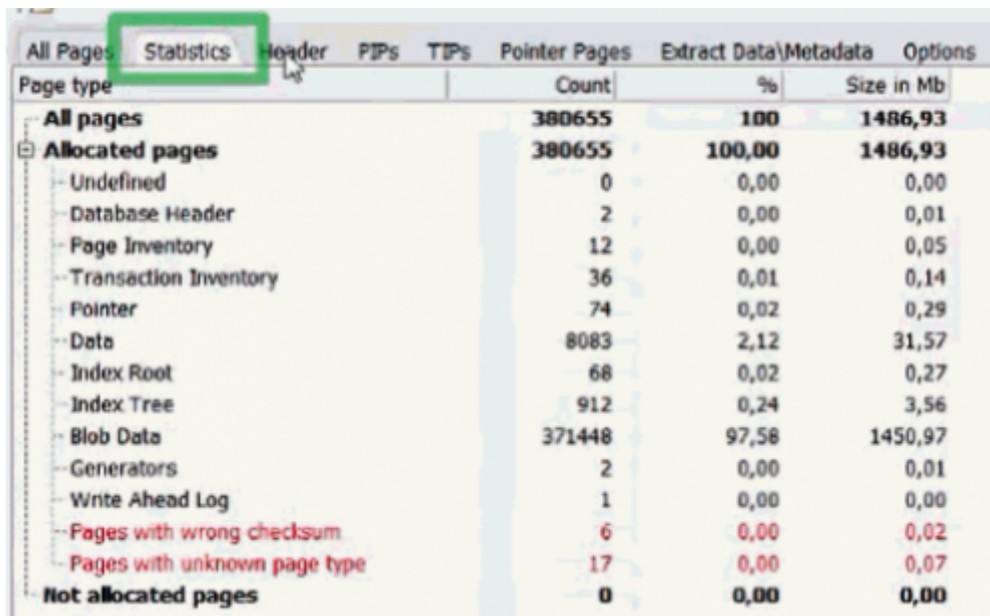
You can read more about data pages and pointer pages in the following articles:

- [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.

All Pages			Statistics	Header	PIPs	T-Is	Pointer Pages	Extract Data/Metadata	Options										
Property	Value		00000000	01	00	39	30	98	DD	08	00	00	00	00	00	00	00	00	00
Page Header			00000010	00	10	0B	00	03	00	00	00	00	00	00	65	DD	08	00	00
Database properties			00000020	66	DD	08	00	8D	DD	08	00	00	00	22	01	4C	D6	00	00
Page size	4096		00000030	74	66	86	26	00	00	00	00	00	00	00	10	00	00	00	00
Version of on-disk structure	11 (Firebird)		00000040	00	00	96	00	40	1F	00	00	01	00	00	00	66	DD	08	00
Page number of RDBSPAGES relation	3		00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Page number of next header page	0		00000060	06	04	20	4E	00	00	03	28	48	3A	5C	44	61	74	65	6E
Oldest interesting transaction	580965		00000070	44	44	5C	53	6D	61	72	74	53	6F	66	74	5C	44	42	5C
Oldest transaction thought active	580966		00000080	43	41	52	64	49	4F	42	61	53	45	5F	32	2E	63	69	62
Next transaction id	581005		00000090	04	04	67	C5	07	00	00	00	00	00	00	00	00	00	00	00
Sequence number of file	0		000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Date of creation	29.01.2009		000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Time of creation	17:57:14		000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Next attachment id	0		000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Event count for shadow synchronization	0		000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Implementation number	16		000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Update version of ODS	0		00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Update version of ODS at creation	0		00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Offset of HDR_end in page	150		00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Page buffers for database cache	8000		00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Bumped transaction id for log optimization	1		00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Oldest snapshot of active transactions	580966		00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Flags			00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Active shadow	☐		00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Forced writes	☒		00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Short-term journaling	☐		00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Long-term journaling	☐		000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Don't calculate checksums	☐		000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Don't reserve space for versions	☒		000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Disable using shared cache file	☐		000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Database is shutdown	☐																		
Database SQL dialect 3	☒																		
Database is read-only	☐																		

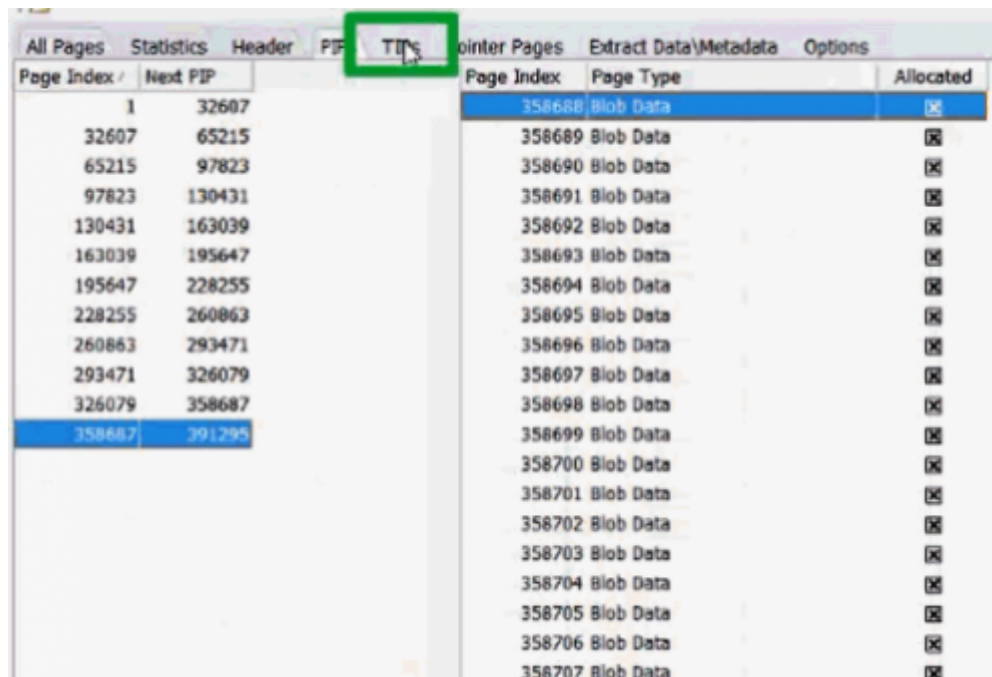
You can read more about PIPs in the following articles:

- [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.



Page Index	Next PIP	Page Index	Page Type	Allocated
1	32607	358688	Blob Data	☒
32607	65215	358689	Blob Data	☒
65215	97823	358690	Blob Data	☒
97823	130431	358691	Blob Data	☒
130431	163039	358692	Blob Data	☒
163039	195647	358693	Blob Data	☒
195647	228255	358694	Blob Data	☒
228255	260863	358695	Blob Data	☒
260863	293471	358696	Blob Data	☒
293471	326079	358697	Blob Data	☒
326079	358687	358698	Blob Data	☒
358687	391295	358699	Blob Data	☒
		358700	Blob Data	☒
		358701	Blob Data	☒
		358702	Blob Data	☒
		358703	Blob Data	☒
		358704	Blob Data	☒
		358705	Blob Data	☒
		358706	Blob Data	☒
		358707	Blob Data	☒

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.

Data/metadata extract options

Extract into: New database
Extract what: Metadata and data of selected tables
Target server version: Firebird 2.5 ☒ SQL Dialect 3

New database options

Database connection string: localhost:C:\employee_repair.fdb
Client library: C:\Program Files (x86)\Firebird\Firebird_2_5\bin\fbclient.dll
User name: SYSDBA
Password: *****
Page size: 4096
Default charset: WIN1252

☒ Use EXECUTE BLOCK for better performance
☒ Use short INSERT (without list of field names) if possible
☒ Drop target database if exists

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.

All Pages	Statistics	Header	PIPs	TIPs	Pointer Pages	Output	Extract Data\Metadata	Errors: 23	Options
Creating procedure SPGETIFMSUR ... Successful.									
Creating procedure SPGETILAKTISUR ... Successful.									
Creating procedure SPGETILBOMSUR ... Successful.									
Creating procedure SPGETILPSURS ... Successful.									
Creating procedure SPGETILZKGSUR ... Successful.									
Creating procedure SPGETIPATIRADATSUR ... Successful.									
Creating procedure SPGETIPATIRKDATSUR ... Successful.									
Creating procedure SPGETISUR ... FAILED!									
Creating procedure SPGETISURMUL ... FAILED!									
Creating procedure SPGETISYSFARSID ... Successful.									
Creating procedure SPGETISYSFARSUR ... Successful.									
Creating procedure SPGETITEXT ... Successful.									
Creating procedure SPGETIUBEKGSUR ... Successful.									
Creating procedure SPGETIUREKGNKSUR ... Successful.									
Creating procedure SPGETIUREKGSUR ... Successful.									
Creating procedure SPGETIURTAUSWSUR ... Successful.									
Creating procedure SPGETIURTRAINASUR ... Successful.									
Creating procedure SPGETIURTRAINSUR ... Successful.									
Creating procedure SPGETIURYKGSUR ... Successful.									
Creating procedure SPGETIURLIST ... Successful.									
Creating procedure SPGETIURLISTOMNEPAT ... Successful.									
Writing triggers...									
Creating trigger TBBUTBPATEREIGNISUPD ... Successful.									
Creating trigger TRAI_TBLEKGPARAUFN_0 ... Successful.									
Creating trigger TRAIUFPARAOFNPATUPD ... FAILED!									
Creating trigger TRAIUTREDAUPDFAT ... Successful.									
Creating trigger TRAILAKTATPARAOFNPATUPD ... Successful.									
Creating trigger TRAILBDMPARAOFNPATUPD ... Successful.									
Creating trigger TRAITBIEREINISUPD ... Successful.									

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 [IBExpert\\$DBINSIDE\\$ERRORS](#) table. This can be opened separately from IBExpert's [Database Explorer](#), once you have registered the repaired database for use in IBExpert.

All Pages Statistics Header PIPs TIPS Pointer Pages Output Extract Data\Metadata Errors: 52 Options				
48 error(s) occurred while executing DML/DOL statements. Erroneous statements and error messages have been stored in IBE\$DBINSIDE\$ERRORS table in order of appearance.				
#	Message	Data page	Record #	Relation ID
6	violation of FOREIGN KEY constraint "".	0		
7	Invalid token.	0		
8	Invalid token.	0		
9	Invalid token.	0		
10	Invalid token.	0		
11	Column does not belong to referenced table.	0		
12	Invalid token.	0		
13	Column does not belong to referenced table.	0		
14	Column does not belong to referenced table.	0		
15	Invalid token.	0		
16	Invalid token.	0		
17	Invalid token.	0		
18	Invalid token.	0		
19	Invalid token.	0		
20	Invalid token.	0		
21	Invalid token.	0		
22	Invalid token.	0		
23	Invalid token.	0		
24	Invalid token.	0		
25	Invalid token.	0		
26	Invalid token.	0		
27	Invalid token.	0		
28	Invalid token.	0		
29	Invalid token.	0		
30	Invalid token.	0		

Error additional info

Invalid token.
invalid request BLR at offset 21.
function UDFNEWSUR_ is not defined.
module name or entrypoint could not be found.

CREATE TRIGGER TRBITBLZKGGHESSURCRUPD_ FOR TBLZKGGHES_
ACTIVE BEFORE INSERT POSITION 0
as
declare variable vcNewSur_ varchar(30);
begin

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:

All Pages Statistics Header PIPs TIPS Pointer Pages Output Extract Data\Metadata Errors: 52 Options
Creating trigger TRAU_TBUEREIGNIS_2... Successful.
Creating trigger TRAU_LAKTATPARAUFNPATUPD_... Successful.
Creating trigger TRAUTBBLOBS_0... FAILED!
Creating trigger TRAUTBIFNPARAUFNPATUPD_... Successful.
Creating trigger TRAUTBLBDMPARAUFN_0... Successful.
Creating trigger TRAUTBLZKGGPARAUFNPDPAT_... Successful.
Creating trigger TRAUTBPATEREIGNISUPDOER_... FAILED!
Creating trigger TRAUTBPATIENTEREIGNIS_... FAILED!
Creating trigger TRAUTBUEREIGNISUPDSIGPARAUSW_... Successful.
Creating trigger TRAUTBUPARAUFNPDPAT_... Successful.
Creating trigger TRAUTBUTRKDATPATUPD_... Successful.
Creating trigger TRBDTBLOBS_0... FAILED!
Creating trigger TRBDTBATEREIGNIS_0... Successful.
Creating trigger TRBI_TBLZKGGPARAUFN_0... FAILED!
Creating trigger TRBI_TBLZKGGPARAUSW_0... FAILED!

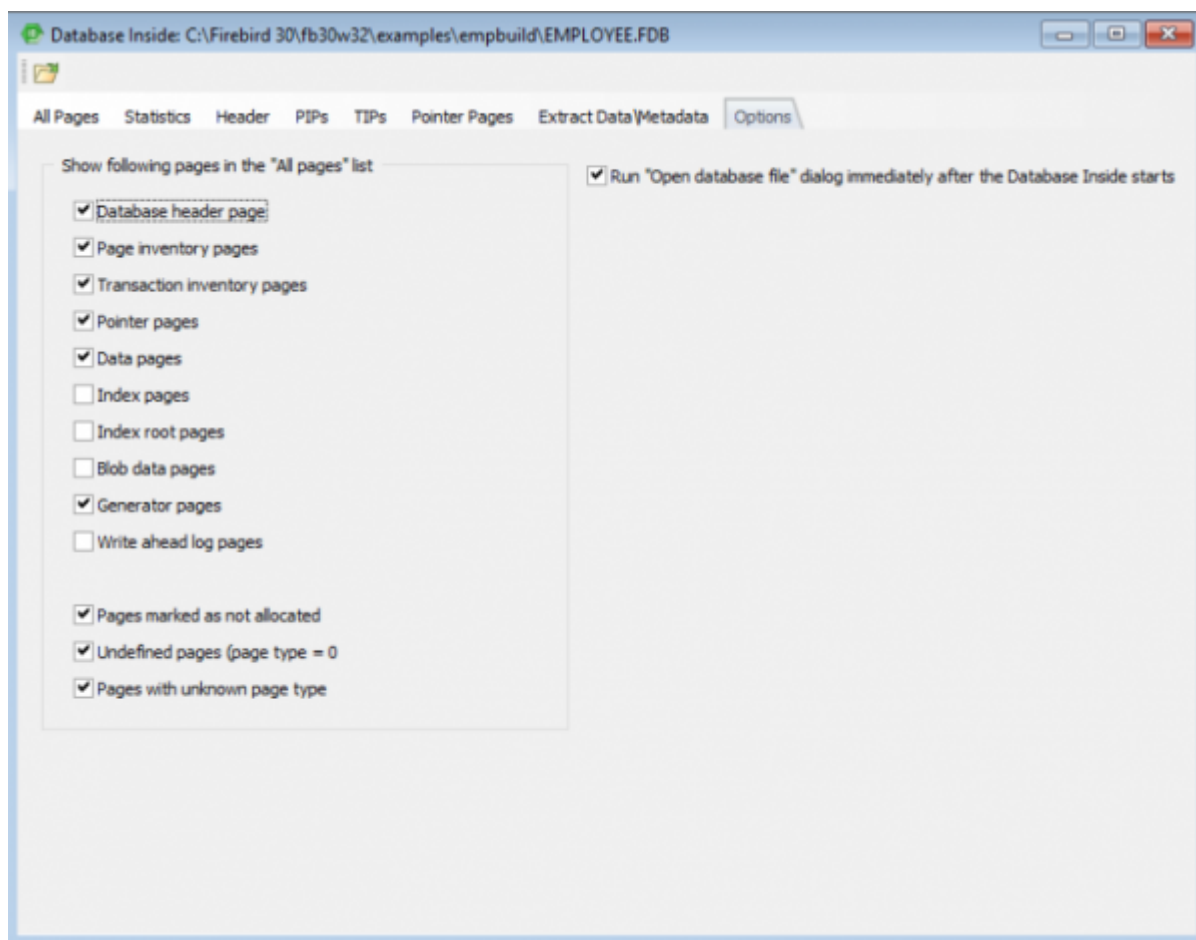
You can then [register the repaired database](#) for use in IBExpert.

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](#)

From:
<https://www.ibexpert.com/docu/> - **IBExpert**

Permanent link:
<https://www.ibexpert.com/docu/doku.php?id=02-ibexpert:02-08-ibexpert-tools-menu:database-inside>

Last update: **2023/10/09 20:10**

