

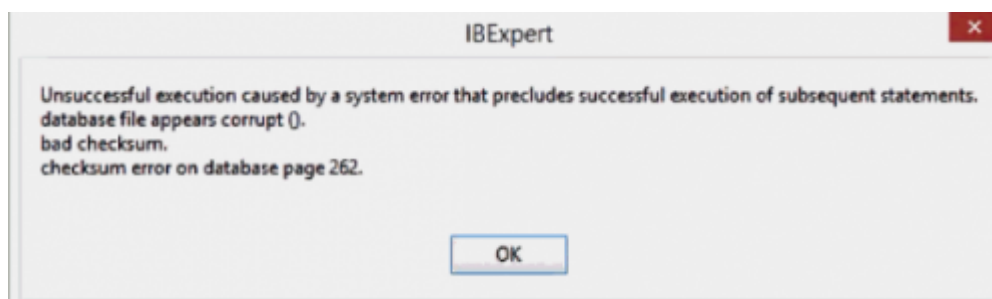
Database Inside

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

IBEExpert 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*:

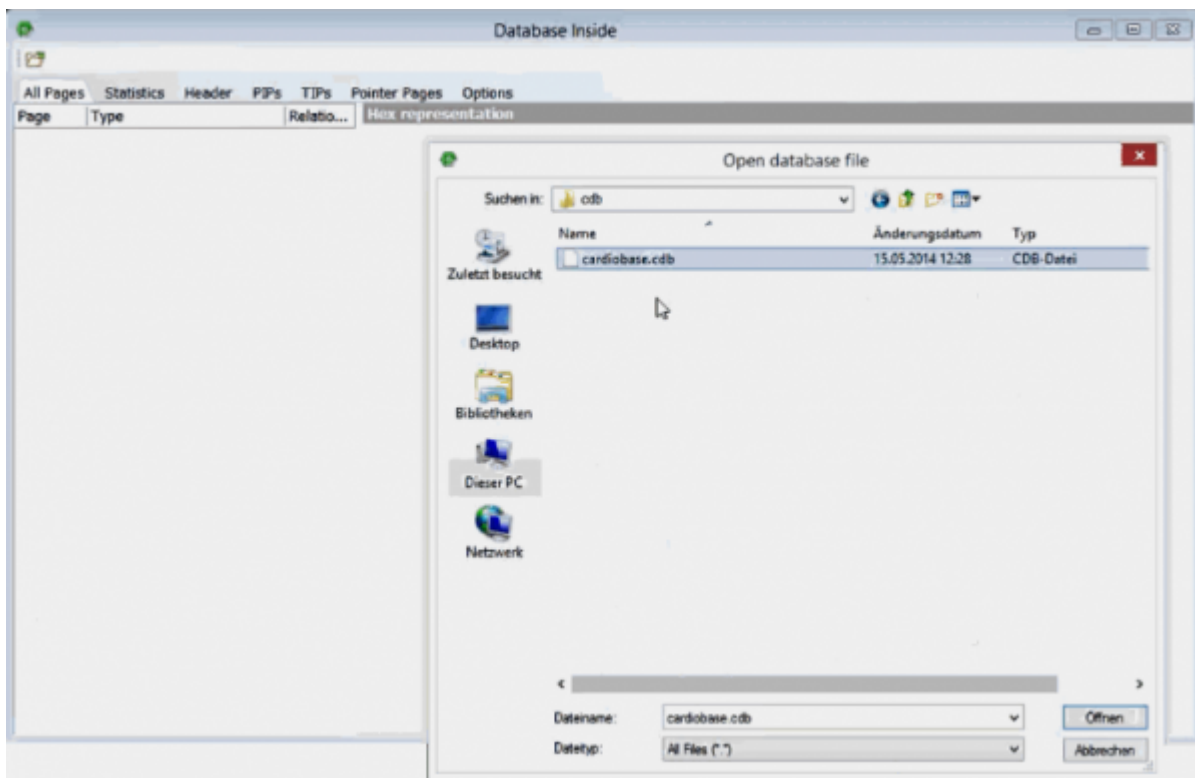


It is not even possible to perform a successful backup – even without the [Garbage Collection](#) or with or without any other options. In this example, one of the system tables is damaged; so it's a pretty serious problem. Even using [gfix](#) to validate the database is not successful.

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:

Last update: 2023/10/09 20:10 02-ibexpert:02-08-ibexpert-tools-menu:database-inside <http://ibexpert.com/docu/doku.php?id=02-ibexpert:02-08-ibexpert-tools-menu:database-inside>



The selected database is opened and loaded accordingly:

All Pages	Statistics	Header	FPs	TPs	Printer Pages	Extract DetailMetadata	Options
Page	Type	Ratio...					
0	Database Header						
1	Page Inventory						
2	Write Ahead Log						
3	Pointer						
4	Index Root	0					
5	Data	0					
6	Pointer	1					
7	Index Root	0					
8	Pointer	2					
9	Index Root	0					
10	Index Root	3					
11	Index Root	0					
12	Pointer	4					
13	Index Root	0					
14	Pointer	5					
15	Index Root	0					
16	Pointer	6					
17	Index Root	0					
18	Pointer	7					
19	Index Root	1					
20	Pointer	8					
21	Index Root	0					
22	Pointer	9					
23	Index Root	0					
24	Pointer	10					
25	Index Root	0					
26	Pointer	11					
27	Index Root	0					
28	Pointer	12					
29	Index Root	0					
30	Pointer	13					
31	Index Root	0					
32	Pointer	14					
33	Index Root	0					
34	Pointer	15					
35	Index Root	0					
36	Pointer	16					
37	Index Root	0					
38	Pointer	17					
39	Index Root	0					
40	Pointer	18					
41	Index Root	0					
42	Pointer	19					
43	Index Root	0					

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*.

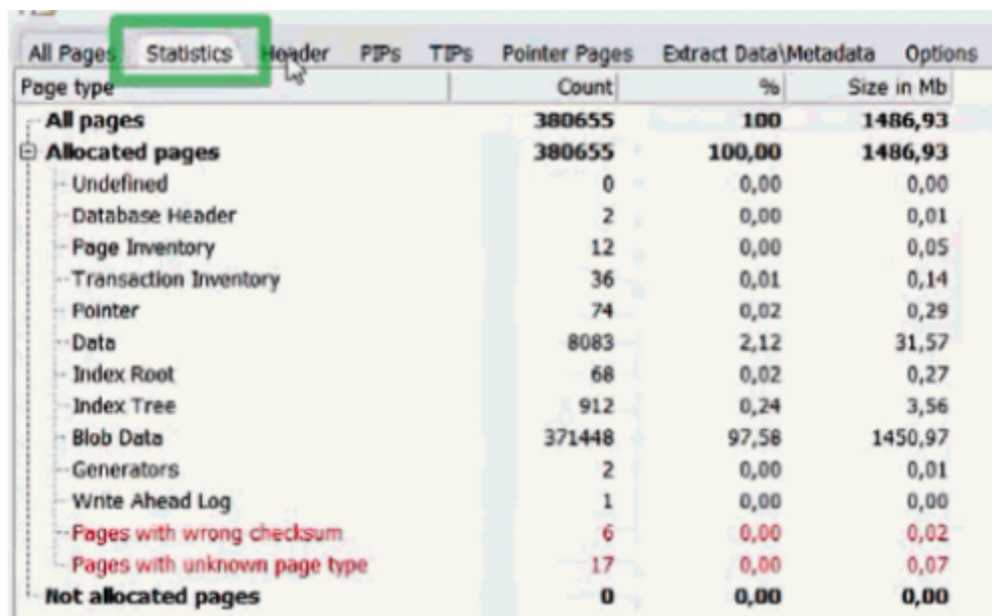
You can read more about data pages and pointer pages in the following articles:

- [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**.



All Pages	Statistics	Header	PIPs	TIPs	Pointer Pages	Extract Data\Metadata	Options
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				

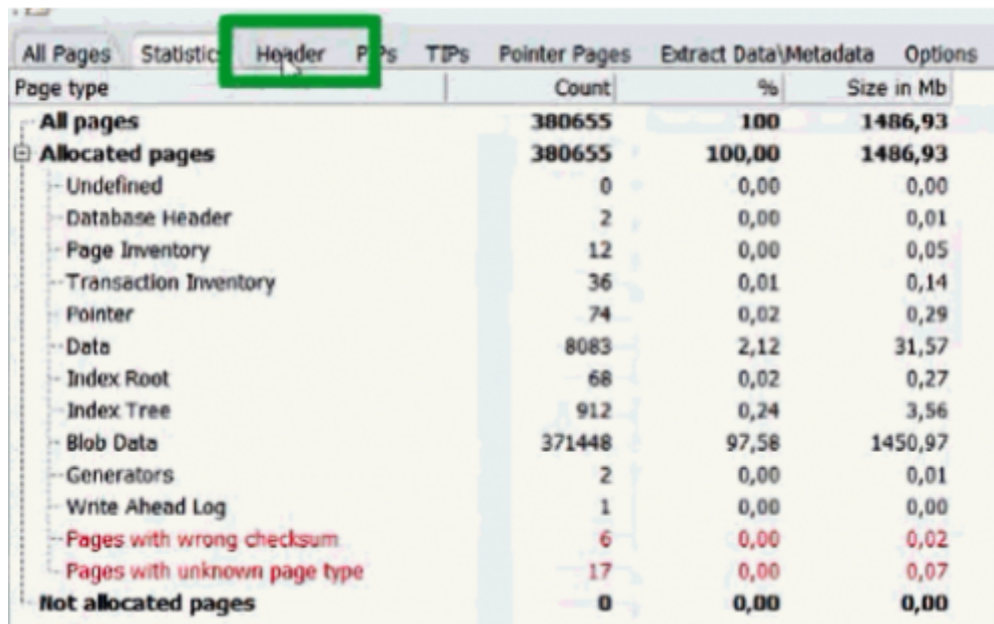
For further information regarding the individual page types, refer to the Database Technology article, [Firebird for the database expert: page types](#) and the IBExpert documentation chapter, [Database Statistics](#).

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	TIPs	Pointer Pages	Extract Data/Metadata	Options
Property		Value					
Page Header							
Database properties							
Page size		4096					
Version of on-disk structure		11 (Firebird)					
Page number of RDBSPAGES relation		3					
Page number of next header page		0					
Oldest interesting transaction		580965					
Oldest transaction thought active		580966					
Next transaction id		581005					
Sequence number of file		0					
Date of creation		29.01.2009					
Time of creation		17:57:14					
Next attachment id		0					
Event count for shadow synchronization		0					
Implementation number		16					
Update version of ODS		0					
Update version of ODS at creation		0					
Offset of HDR_end in page		150					
Page buffers for database cache		8000					
Bumped transaction id for log optimization		1					
Oldest snapshot of active transactions		580966					
Flags							
Active shadow		<input type="checkbox"/>					
Forced writes		<input checked="" type="checkbox"/>					
Short-term journaling		<input type="checkbox"/>					
Long-term journaling		<input type="checkbox"/>					
Don't calculate checksums		<input type="checkbox"/>					
Don't reserve space for versions		<input checked="" type="checkbox"/>					
Disable using shared cache file		<input type="checkbox"/>					
Database is shutdown		<input type="checkbox"/>					
Database SQL dialect 3		<input checked="" type="checkbox"/>					
Database is read-only		<input type="checkbox"/>					

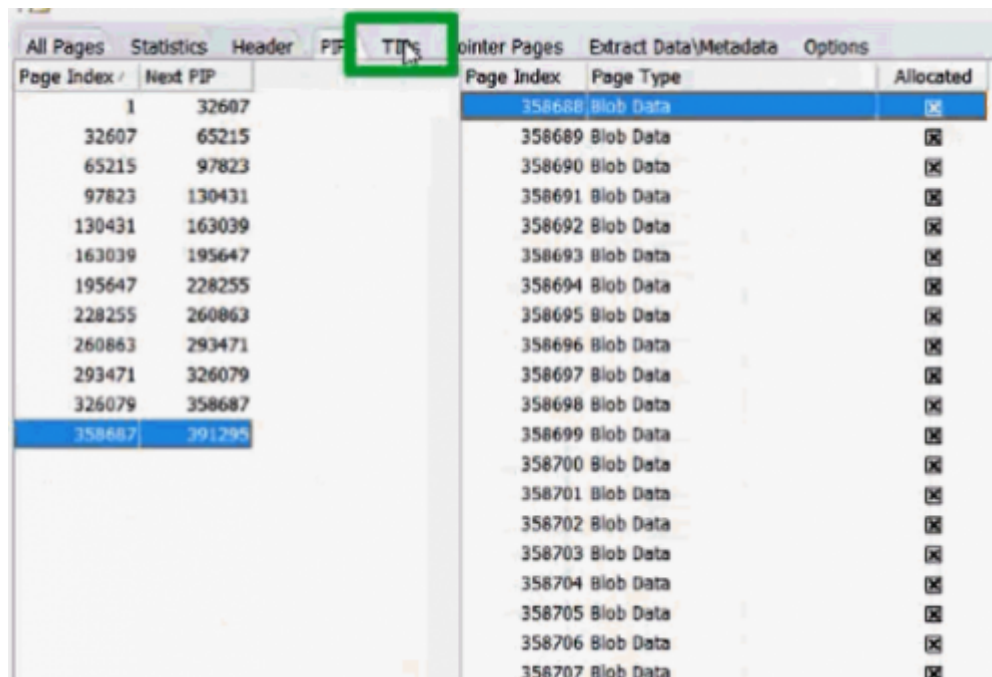
You can read more about PIPs in the following articles:

- [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	<input checked="" type="checkbox"/>
32607	65215	358689	Blob Data	<input checked="" type="checkbox"/>
65215	97823	358690	Blob Data	<input checked="" type="checkbox"/>
97823	130431	358691	Blob Data	<input checked="" type="checkbox"/>
130431	163039	358692	Blob Data	<input checked="" type="checkbox"/>
163039	195647	358693	Blob Data	<input checked="" type="checkbox"/>
195647	228255	358694	Blob Data	<input checked="" type="checkbox"/>
228255	260863	358695	Blob Data	<input checked="" type="checkbox"/>
260863	293471	358696	Blob Data	<input checked="" type="checkbox"/>
293471	326079	358697	Blob Data	<input checked="" type="checkbox"/>
326079	358687	358698	Blob Data	<input checked="" type="checkbox"/>
358687	391295	358699	Blob Data	<input checked="" type="checkbox"/>
		358700	Blob Data	<input checked="" type="checkbox"/>
		358701	Blob Data	<input checked="" type="checkbox"/>
		358702	Blob Data	<input checked="" type="checkbox"/>
		358703	Blob Data	<input checked="" type="checkbox"/>
		358704	Blob Data	<input checked="" type="checkbox"/>
		358705	Blob Data	<input checked="" type="checkbox"/>
		358706	Blob Data	<input checked="" type="checkbox"/>
		358707	Blob Data	<input checked="" type="checkbox"/>

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.

The screenshot shows the 'Data/metadata extract options' dialog box. It has two main sections: 'Data/metadata extract options' and 'New database options'. In the first section, 'Extract into' is set to 'New database' and 'Extract what' is set to 'Metadata and data of selected tables'. 'Target server version' is 'Firebird 2.5' and 'SQL Dialect 3' is checked. The second section, 'New database options', contains a 'Database connection string' field with 'localhost:C:\employee_repair.fdb', a 'Client library' field with 'C:\Program Files (x86)\Firebird\Firebird_2_5\bin\fbclient.dll', and fields for 'User name' (SYSDBA), 'Password' (masked with asterisks), 'Page size' (4096), and 'Default charset' (WIN1252). At the bottom, three checkboxes are checked: 'Use EXECUTE BLOCK for better performance', 'Use short INSERT (without list of field names) if possible', and '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	SPGETPATRADATSUR ...	Successful.							
Creating procedure	SPGETPATRKDATSUR ...	Successful.							
Creating procedure	SPGETSUR ...	FAILED!							
Creating procedure	SPGETSURMUL ...	FAILED!							
Creating procedure	SPGETSYSFARSID ...	Successful.							
Creating procedure	SPGETSYSFARSUR ...	Successful.							
Creating procedure	SPGETTEXT ...	Successful.							
Creating procedure	SPGETUBEKGSUR ...	Successful.							
Creating procedure	SPGETUREKGNKSUR ...	Successful.							
Creating procedure	SPGETUREKGSUR ...	Successful.							
Creating procedure	SPGETURTAUSWSUR ...	Successful.							
Creating procedure	SPGETURTRAINASUR ...	Successful.							
Creating procedure	SPGETURTRAINSUR ...	Successful.							
Creating procedure	SPGETURYKGSUR ...	Successful.							
Creating procedure	SPGETUTRLIST ...	Successful.							
Creating procedure	SPGETUTRLISTOMNEPAT ...	Successful.							
Writing triggers...									
Creating trigger	TBBUTBPATEREIGNISUPD ...	Successful.							
Creating trigger	TRAI_TBLZEKGPARAUFN_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/DDL statements. Erroneous statements and error messages have been stored in IBESDBINSIDE\$ERRORS table in order of appearance.									
#	Message	Data page	Record #	Relation ID	Column #				
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 TRBITBLZKNGHRESSURCRUPD_ FOR TBLZKNGHES_
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_TBUEIREIGNIS_2...	Successful.							
Creating trigger	TRAULAKTATPARAUFNPATUPD...	Successful.							
Creating trigger	TRAUTBBLOBS_0...	FAILED!							
Creating trigger	TRAUTBIENPARAUFNPATUPD...	Successful.							
Creating trigger	TRAUTBLBDMPARAUFN_0...	Successful.							
Creating trigger	TRAUTBLZKGPARAUFNUPDPAT...	Successful.							
Creating trigger	TRAUTBPATEREIGNISUPDOER...	FAILED!							
Creating trigger	TRAUTBPATIENTEREIGNIS...	FAILED!							
Creating trigger	TRAUTBUEREIGNISUPDSIGPARAUSW...	Successful.							
Creating trigger	TRAUTBUFARAUFNUPDPAT...	Successful.							
Creating trigger	TRAUTBUIRKDATPATUPD...	Successful.							
Creating trigger	TRBDTBLOBS_0...	FAILED!							
Creating trigger	TRBDTPATEREIGNIS_0...	Successful.							
Creating trigger	TRBI_TBLZKGPARAUFN_0...	FAILED!							
Creating trigger	TRBI_TBLZKGPARAUSW_0...	FAILED!							

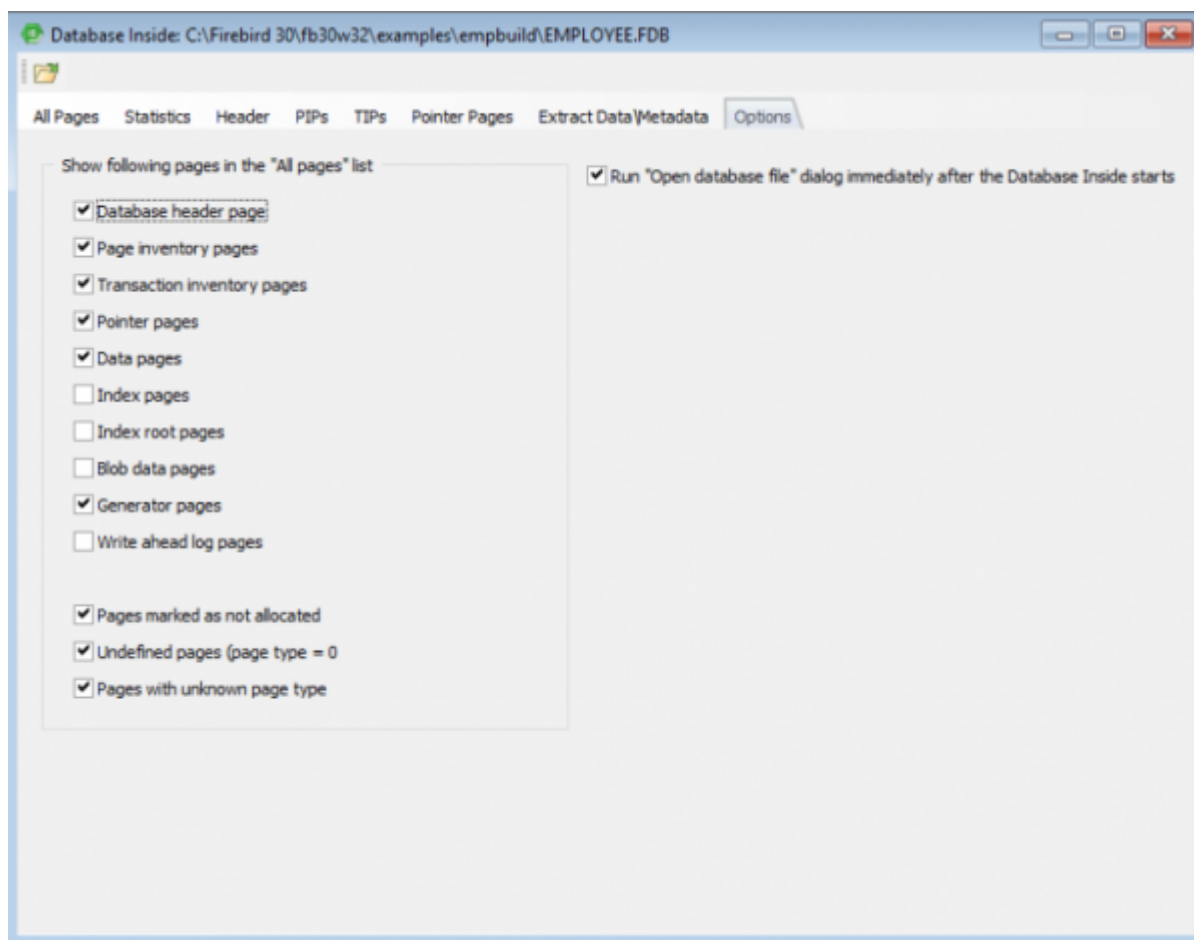
You can then [register the repaired database](#) for use in IBExpert.

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:
<http://ibexpert.com/docu/> - **IBExpert**

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

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

