

Import & export using IBExpert (2)

The following article is intended to give readers an overview of the various methods of importing and exporting data and files to and from a Firebird/InterBase® using the IBExpert Developer Studio.

Exporting [data](#) and/or [metadata](#) is to format data in such a way that it can be used by another application. An application that can export data can create a file in a format that another application understands, enabling the two programs to share the same data.

The flip side of exporting is importing. Importing refers to the ability of an application to read and use data and/or metadata produced by a different application. Exporting implies that the sending application reformats the data for the receiving application, whereas importing implies that the receiving application does the reformatting.

Importing and exporting data in Firebird

The Firebird core only offers import and export using external files, which requires a setting in and restarting the server.

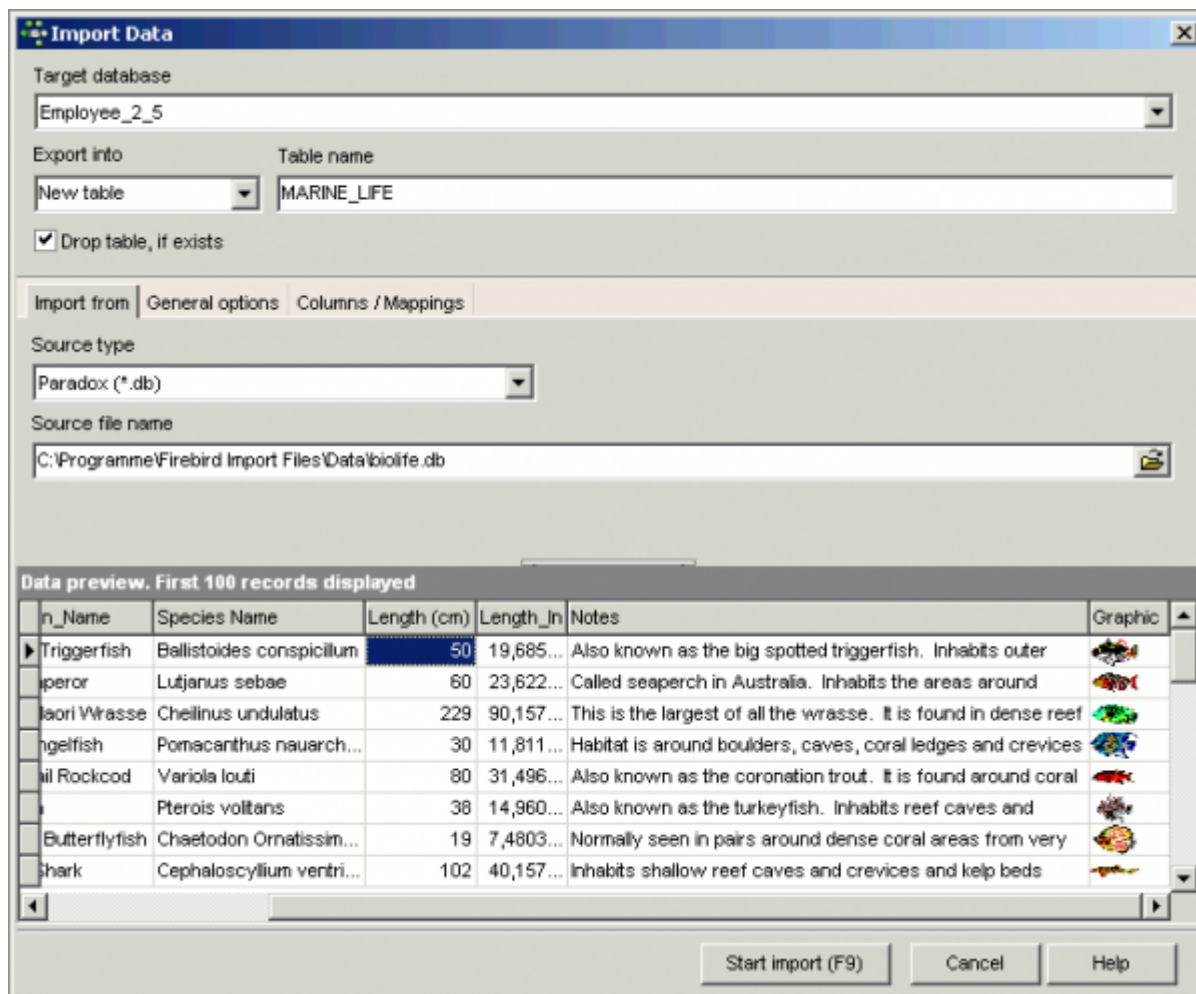
The files can be defined by declaring a [table](#):

```
create table external file
```

This function is extremely quick; 100,000 data sets can be imported or exported every second. It is however limited for certain [datatypes](#), particularly those of a variable length, such as [blobs](#). The best solution is to define the table using the above instruction, and defining as far as possible all fields as [CHAR](#).

You can alternatively use the Firebird ODBC driver with any ODBC-capable tool or use one of the various options offered by IBExpert.

Importing data and files using the IBExpert Import data wizard



This feature enables you to quickly and easily import tables and data from a wide range of file formats including the following:

- **Documents**
 - Text document
 - Windows clipboard
- **Spreadsheets / Workbooks**
 - MS Excel (*.xls)
 - Lotus 1-2-3 (*.wk1)
 - Quattro Pro (*.wq1)
 - OpenOffice spreadsheet (*.odt)
- **Databases**
 - Paradox (*.db)
 - dBASE (*.dbf)
 - MS Access (*.mdb)
 - Advantage table (*.adt)
 - DBISAM table (*.dat)
 - Clarion table (*.dat)
- **Miscellaneous**
 - HTML file (*.htm, *.html)
 - XML file (*.xml)
 - Windows address book (*.wab)
 - VCalender Data file (*.vcs)

Further features include specification of date/time appearance, and you can easily adjust column

names, datatypes and mappings, previewing your data content on screen before importing it.

This feature is described in detail in the [IBExpert documentation](#) chapter, [Import data](#).

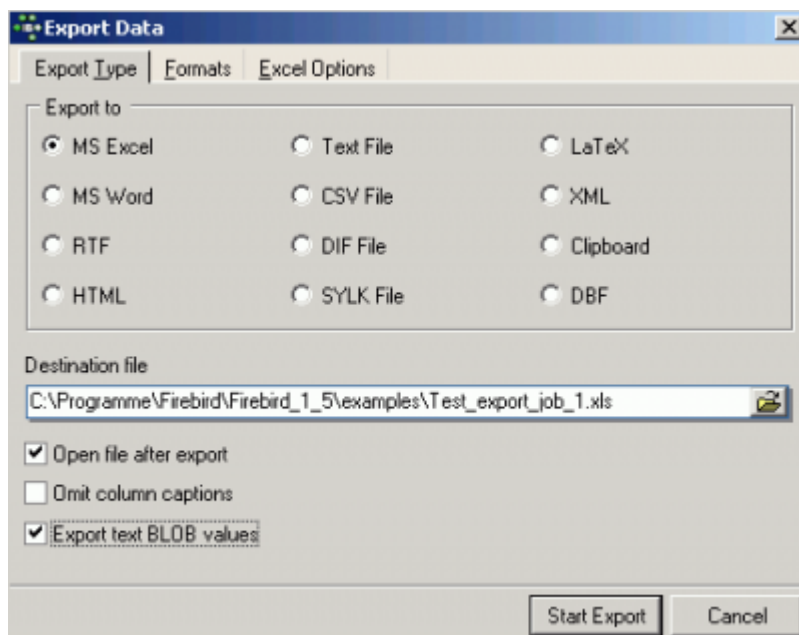
Exporting table data and query result sets from the IBExpert editors

The quickest method to export table data is from the [Data page](#) in the [Table Editor](#) and [View Editor](#), and query result sets using the [Results page](#) in the [SQL Editor](#) by simply clicking the



icon or using the key combination [Ctrl + E] to open the Data Export window.

The first page in the Export Data dialog, Export Type, offers a wide range of formats, including Excel, MS Word, RTF, HTML, Text, CSV, DIF, SYLK, LaTeX, SML, Clipboard and DBF, which can be simply and quickly specified per mouse click (or using the directional keys).

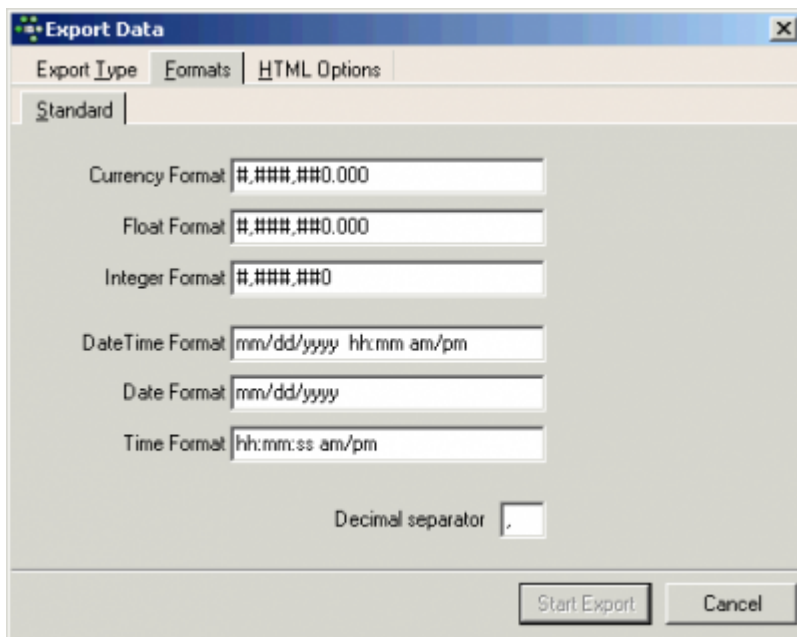


The destination file name must also be specified, and check options allow you to define whether the resulting export file should be opened following the data export or not, and - for certain export formats - whether column headings should be omitted or not, and whether text blob values should also be exported.

Should you encounter problems when exporting text blob values, please check that the Show text blobs as memo option is checked on the Grid page found under the IBExpert menu item [Options / Environment Options](#).

Depending on the format, further options can be specified on the second and third pages, Formats

and Options, specific to the export type. The Formats page is available for all export types, with the exception of XML.



The screenshot shows the 'Export Data' dialog box with the 'Formats' tab selected. The 'Standard' export type is chosen. The following format fields are visible:

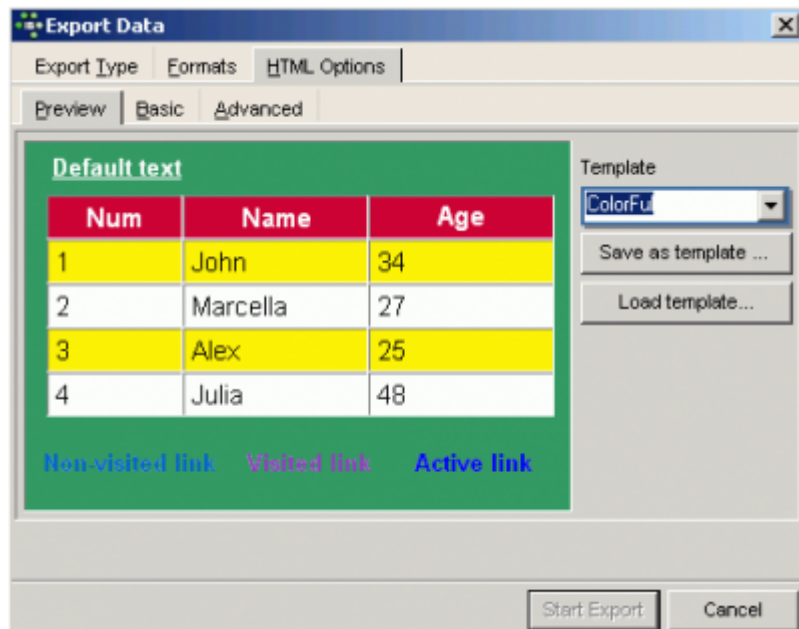
- Currency Format: #,###,##0.000
- Float Format: #,###,##0.000
- Integer Format: #,###,##0
- DateTime Format: mm/dd/yyyy hh:mm am/pm
- Date Format: mm/dd/yyyy
- Time Format: hh:mm:ss am/pm

The 'Decimal separator' is set to a comma (,). At the bottom, there are 'Start Export' and 'Cancel' buttons.

Here it is possible to specify a range of numerical formats, including currency, float, integer, date, time or date and time, as well as the decimal separator. Please note that not all of these options may be altered for all export types (for example when exporting to DBF it is only possible to specify the formats for date/time and time).

Depending upon which format has been specified, additional options may be offered on the third page, for example:

- **Excel:** specification of page header and footer.
- **HTML:** template selection and preview, title, header and footer text as well as a wide range of advanced options.
- **CSV:** Quote String check option, and user specification of CSV separator.
- **XML:** Encoding format may be selected from a pull-down list. There are also check options to export String, Memo and DateTime fields as text.
- **DBF:** check options to export strings to DOS, long strings to Memo, and to extract DateTime as Date.



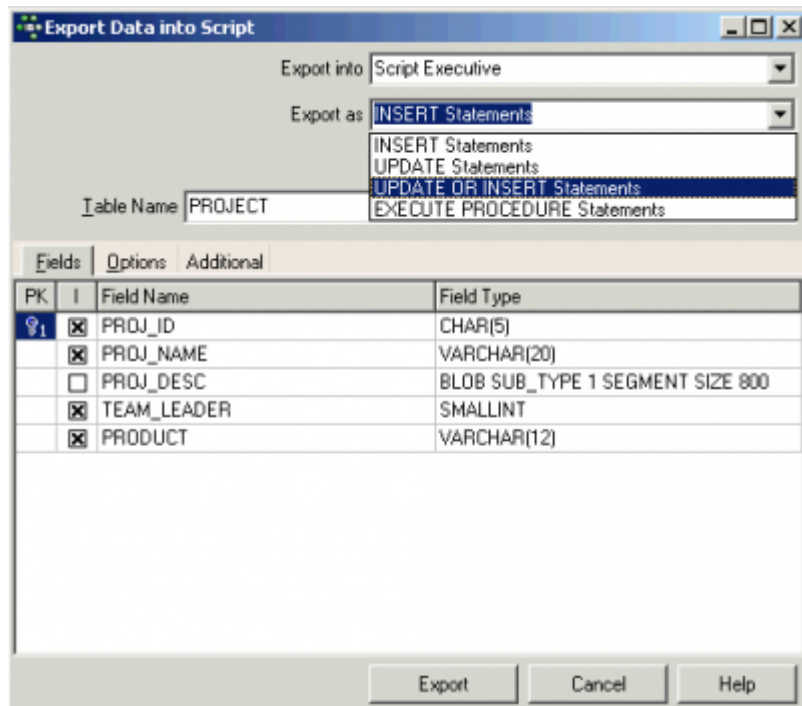
The export is then finally started using the Start Export button in the bottom right-hand corner. Following a successful export, a message appears informing of the total number of records exported.

Export data into script

Using the right-hand icon in the [Table Editor](#) and [View Editor](#) or the [Results page](#) in the [SQL Editor](#) (Export data into script):



the data can be exported into an insert SQL script (without the blob fields).

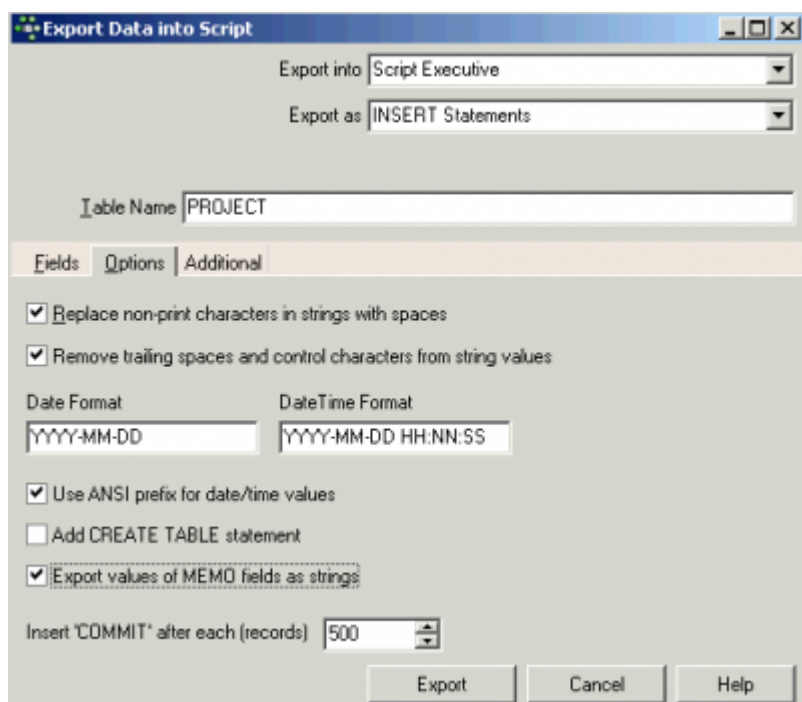


The following options may be selected before starting the export:

- **Export into:** File, Clipboard or Script Executive.
- **Export as:** INSERT statements, UPDATE statements, UPDATE OR INSERT statements or as a set of EXECUTE PROCEDURE statements.

Specify the file name if exporting to file and the table name from which the data is to be exported. The Fields page allows the table fields to be selected.

The Options page:



offers a number of options including replacement of non-print characters in strings with spaces, removal of trailing spaces and control characters from string values, date and time specification, whether the CREATE TABLE statement should be added into the script and whether the values of

MEMO fields should be exported as strings or not. The Insert 'COMMIT' after each x records default value of 500 records can be manually altered if wished.

The Additional page allows additional definitions for query to be made, for example, ORDER BY or WHERE clauses.

After completing all specifications as wished, simply click the Export button to perform the data export.

IBExpert can work with scripts larger than 2 GB, you may however prefer to split such large scripts into two or more smaller scripts. This can be done using the [IBExpert Tools menu](#) item, [Extract Metadata](#), where it is possible to specify the option separate files and the maximum file size limit.

For information regarding the ODBC Viewer's Export data into a table functionality, please refer to [below].

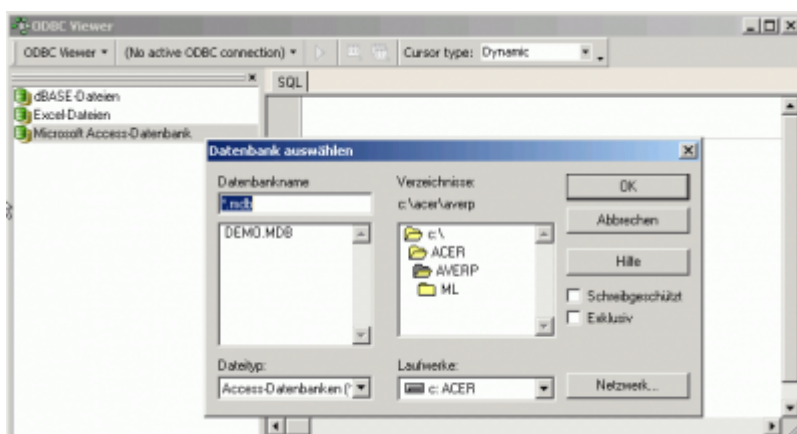
Using IBExpert's ODBC Viewer

The ODBC Viewer allows you to browse data from any ODBC source available on your PC and also import data from an [ODBC](#) source into an SQL script or directly into a Firebird/InterBase® database.

Setting up and testing the ODBC driver

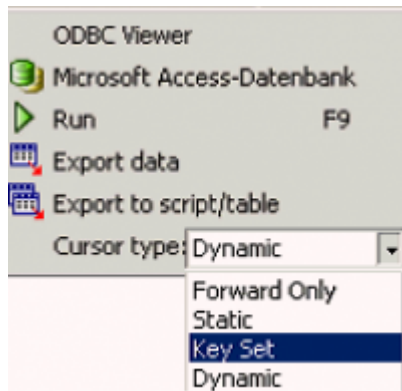
If you need an ODBC driver, it can be downloaded from <https://www.firebirdsql.org>. Then use the Windows menu: Settings / System Control / Administration / Data Source and select fbodbc. This now allows you to access Firebird data from non-Firebird applications such as, for example, OpenOffice Base.

Should you wish to import data from other data sources, please refer to the [IBEBlock](#) chapter, [ODBC access](#)



Using the [IBExpert Tools menu](#) item, [ODBC viewer](#), select the database from the selection of formats: dBASE or Excel files, or Microsoft Access databases, to load the database tables.

The ODBC Viewer toolbar offers the following options:

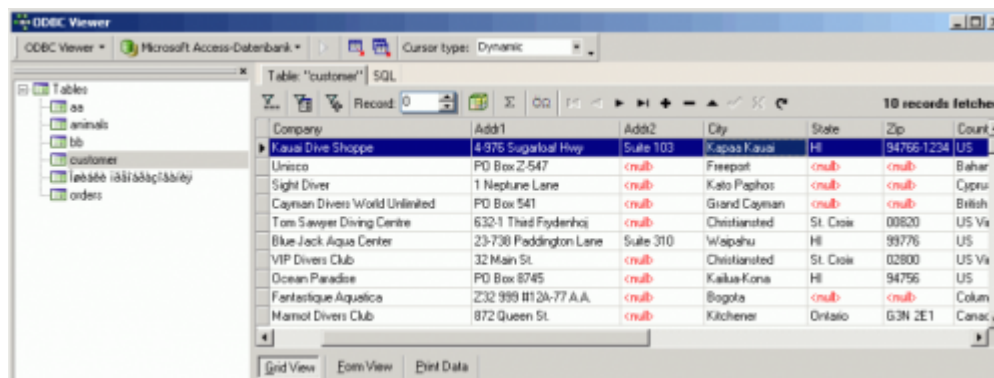


1. Display the active database connection type
2. Run [F9] - runs an SQL
3. [Export data](#)
4. [Exports data to script/file](#)
5. Cursor type: here a choice is offered of Forward Only, Static, Key Set, Dynamic.

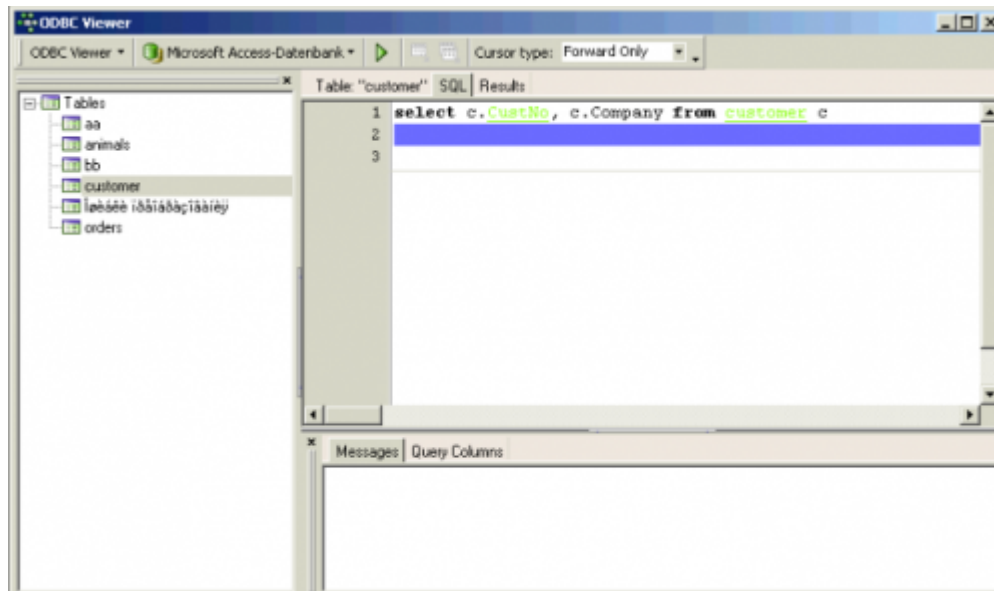
The navigational buttons and icons displayed on the tool bar running across the head of the table data are explained in detail under: [Table Editor / Data Grid](#). The ODBC Viewer's right-click menu is also detailed in this chapter.

The IBExpert [Blob Viewer/Editor](#) is automatically opened by double-clicking on any [blob](#) field.

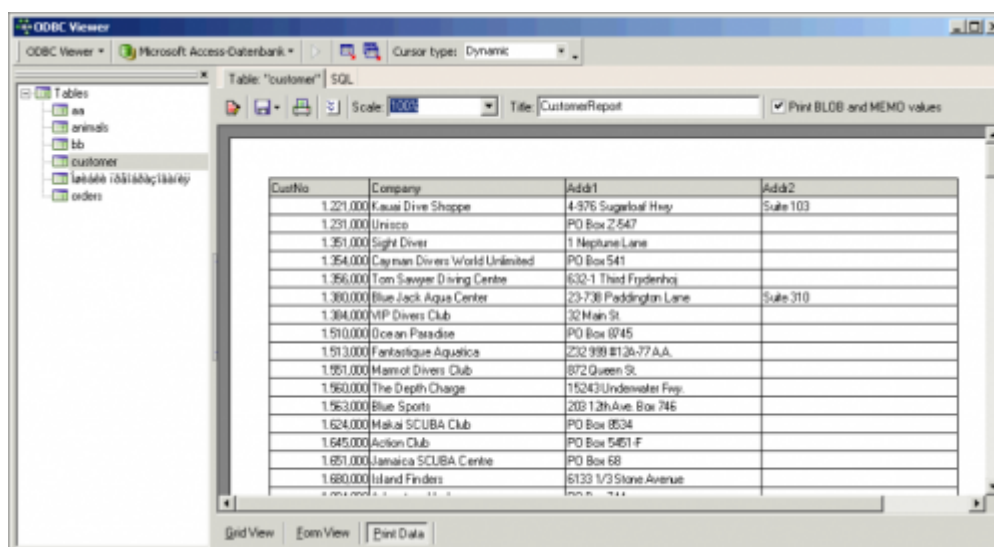
Double-click on a table in the list on the left, to view the data contents. The view type can be easily altered by clicking on the buttons at the bottom left: [Grid View](#), [Form View](#) and there is even the possibility to [print](#) the data. More information regarding these options can be found in the [IBExpert Tools](#) menu item, [SQL Editor / Results](#) and [Table Editor / Data](#).



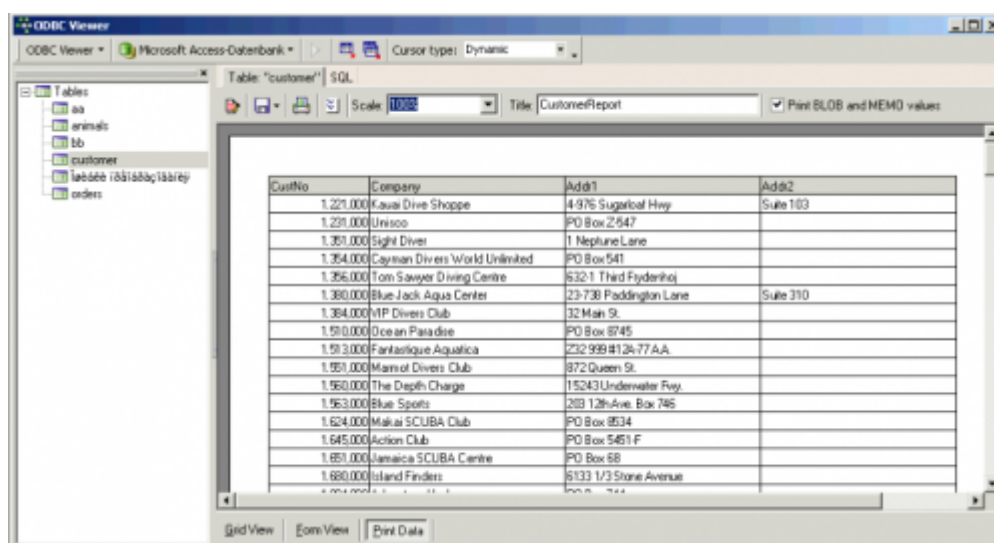
You can even query the table contents:



You can even query the table contents:



and view, print or export the results.

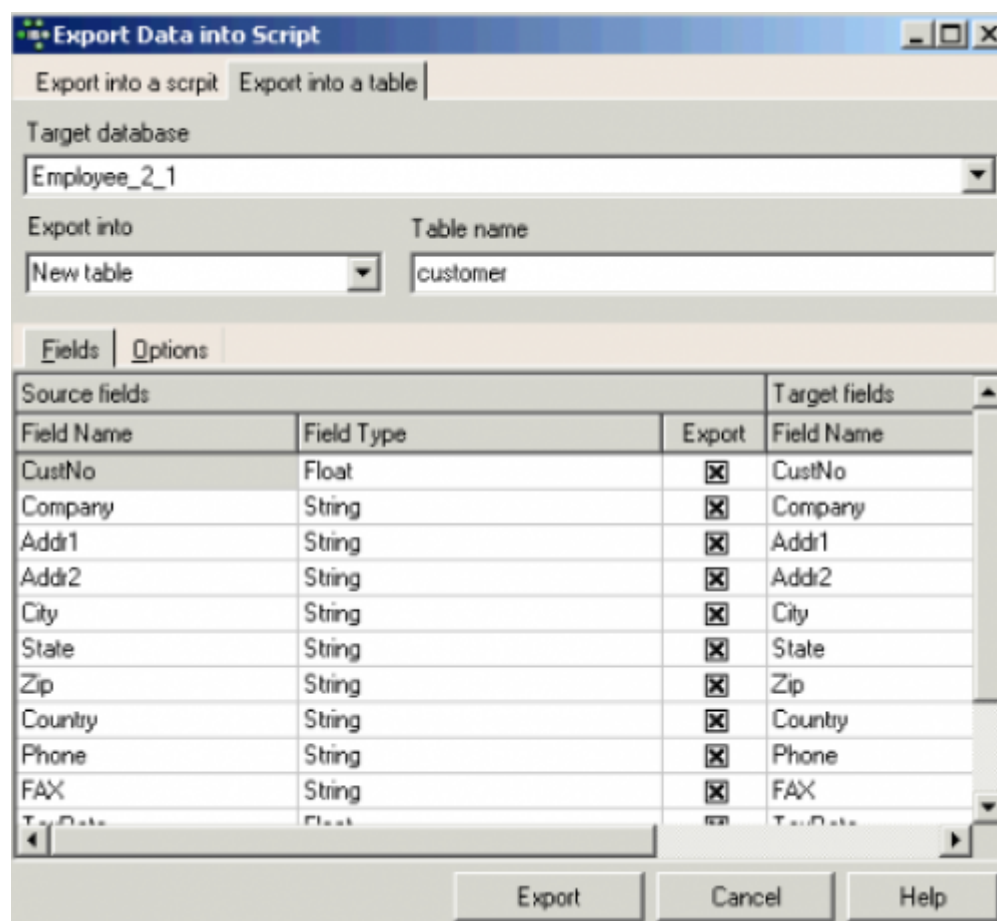


Please refer to [Print Data](#) and [Exporting table data and query result sets from the IBE expert editors](#) above for further information.

Export data into a script

The ODBC Viewer's Export functions are similar to that of the Table Editor's [Export Data into Script](#). However, whereas the Table Editor offers the pages, Fields, Options and Additional, the ODBC Viewer offers the options to Export into script and Export into table, both offering Fields and Options pages.

Export data into a table



The screenshot shows the 'Export Data into Script' dialog box with the 'Export into a table' tab selected. The 'Target database' is 'Employee_2_1'. The 'Export into' dropdown is set to 'New table'. The 'Table name' is 'customer'. The 'Fields' tab is active, showing a list of source fields and target fields. All source fields are checked for export.

Source fields	Field Name	Field Type	Export	Target fields	Field Name
	CustNo	Float	<input checked="" type="checkbox"/>		CustNo
	Company	String	<input checked="" type="checkbox"/>		Company
	Addr1	String	<input checked="" type="checkbox"/>		Addr1
	Addr2	String	<input checked="" type="checkbox"/>		Addr2
	City	String	<input checked="" type="checkbox"/>		City
	State	String	<input checked="" type="checkbox"/>		State
	Zip	String	<input checked="" type="checkbox"/>		Zip
	Country	String	<input checked="" type="checkbox"/>		Country
	Phone	String	<input checked="" type="checkbox"/>		Phone
	FAX	String	<input checked="" type="checkbox"/>		FAX
	TranDate	Float	<input checked="" type="checkbox"/>		TranDate

When exporting into a table, first select the target database for the export, and whether you wish to export into a new or existing table. When existing table is selected, the drop-down list on the right offers a list of all current tables in the selected database.

On the Fields page below, all field names and field types of the source table can be viewed on the left-hand side. All fields are marked as default for exporting. This can be manually altered if wished by clicking or using the space bar on the check boxes. On the right, IBExpert suggests Field Names and Field Types for the new target table. These can be simply overwritten if required.

The Options page offers the Commit after specified number of records option.

Importing Excel files and spread sheets

In Excel it is possible to define a specific area (a whole table or just parts of the data contents) and give this marked area a name (in the upper left area):

	A	B	C	D	E	F	G	H	I	J	K	L
1	CUST_NO	CUSTOMER	CONTACT_FIRST	CONTACT_LAST	PHONE_NO	ADDRESS_LINE1	ADDRESS_LINE2	CITY	STATE_PROVINCE	COUNTRY	POSTAL_CODE	COUNTRY_H
2	1.001	Signature	Dale J.	Little	(619) 530-271	15500 Pacific Heig		San Diego	CA	USA	92121	
3	1.002	Dallas	Te Glen	Brown	(214) 960-223	P. O. Box 47000		Dallas	TX	USA	75205	*
4	1.003	Buttle, G	James	Buttle	(617) 488-186	2300 Newbury Stre	Suite 101	Boston	MA	USA	02115	
5	1.004	Central Bank	Elizabeth	Brocket	61 211 99 88	66 Lloyd Street		Manchester		England	M2 3LA	
6	1.005	DT Systems	Tai	Wu	(852) 850 43	400 Connaught Ro		Central H		Hong Ki		
7	1.006	DataServ	Tomas	Bright	(613) 229 332	2000 Carling Avenu	Suite 150	Ottawa	ON	Canada	K1V 9G1	
8	1.007	Mrs. Bea		Mrs. Beauvais		P. O. Box 22743		Pebble B	CA	USA	93953	
9	1.008	Anini Vac	Leilani	Briggs	(808) 835-760	3320 Lawai Road		Lihue	HI	USA	96766	
10	1.009	Max	Max	Smith	22 01 23	1 Emerald Cove		Turtle Isla		Fiji		*
11	1.010	MPM Cor	Miwako	Miyamoto	3 880 77 19	2-64-7 Sasazuka		Tokyo		Japan	150	
12	1.011	Dynamic	Victor	Granges	01 221 16 50	Florhofgasse 10		Zurich		Switzerl	8005	
13	1.012	3D-Pad	Michelle	Roche	1 43 60 61	22 Place de la Con		Paris		France	75008	
14	1.013	Lorenzi	E Andreas	Lorenzi	02 404 6284	Via Eugenia, 15		Milan		Italy	20124	
15	1.014	Dyno Cor	Greta	Hessels	02 500 5940	Rue Royale 350		Brussels		Belgium	1210	
16	1.015	GeoTech	K.M.	Neppelenbroek	(070) 44 91 16	P.O. Box 702		Den Haag		Netherlands	2514	
17	1.213	Another Test	Debra	Miles	020 456 789	Small Street		Smalltown	CA	USA	92121	

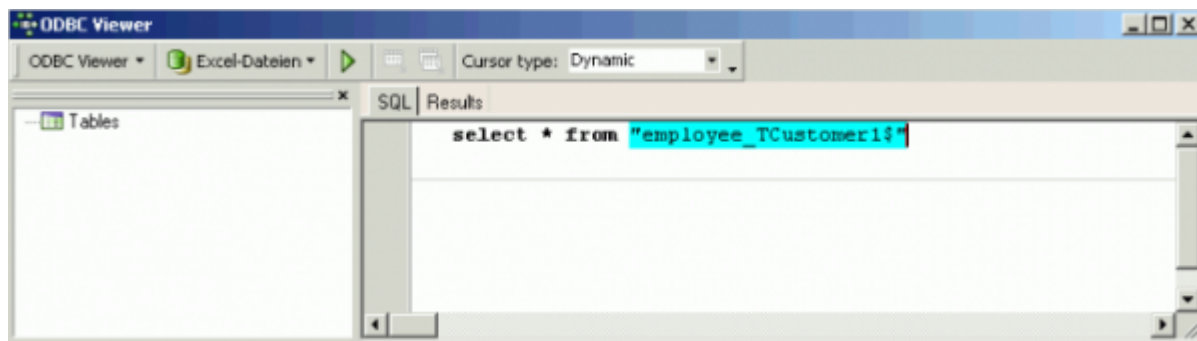
This defined data can then be used as a table in the ODBC Viewer (our example has been defined in Excel as TBL):

CUST_NO	CUSTOMER	CONTACT_FIRST	CONTACT_LAST	PHONE_NO	ADDRESS_LINE1	ADDRESS_LINE2
1.001.000	Signature	Dale J.	Little	(619)	15500 Pacific	
1.002.000	Dallas	Glen	Brown	(214)	P. O. Box 47000	
1.003.000	Buttle, Griffith	James	Buttle	(617)	2300 Newbury	Suite 101
1.004.000	Central Bank	Elizabeth	Brocket	61 211 99	66 Lloyd Street	
1.005.000	DT Systems,	Tai	Wu	(852) 850 43	400 Connaught	
1.006.000	DataServe	Tomas	Bright	(613) 229	2000 Carling	Suite 150
1.007.000	Mrs.		Mrs. Beauvais		P. O. Box 22743	
1.008.000	Anini	Leilani	Briggs	(808)	3320 Lawai Road	
1.009.000	Max	Max	Smith	22 01 23	1 Emerald Cove	
1.010.000	MPM	Miwako	Miyamoto	3 880 77 19	2-64-7 Sasazuka	
1.011.000	Dynamic	Victor	Granges	01 221 16	Florhofgasse 10	
1.012.000	3D-Pad	Michelle	Roche	1 43 60 61	22 Place de la	
1.013.000	Lorenzi	Andreas	Lorenzi	02 404 6284	Via Eugenia, 15	
1.014.000	Dyno	Greta	Hessels	02 500 5940	Rue Royale 350	
1.015.000	GeoTech	K.M.	Neppelenbroek	(070) 44 91	P.O. Box 702	
1.213.000	Another Test	Debra	Miles	020 456 789	Small Street	

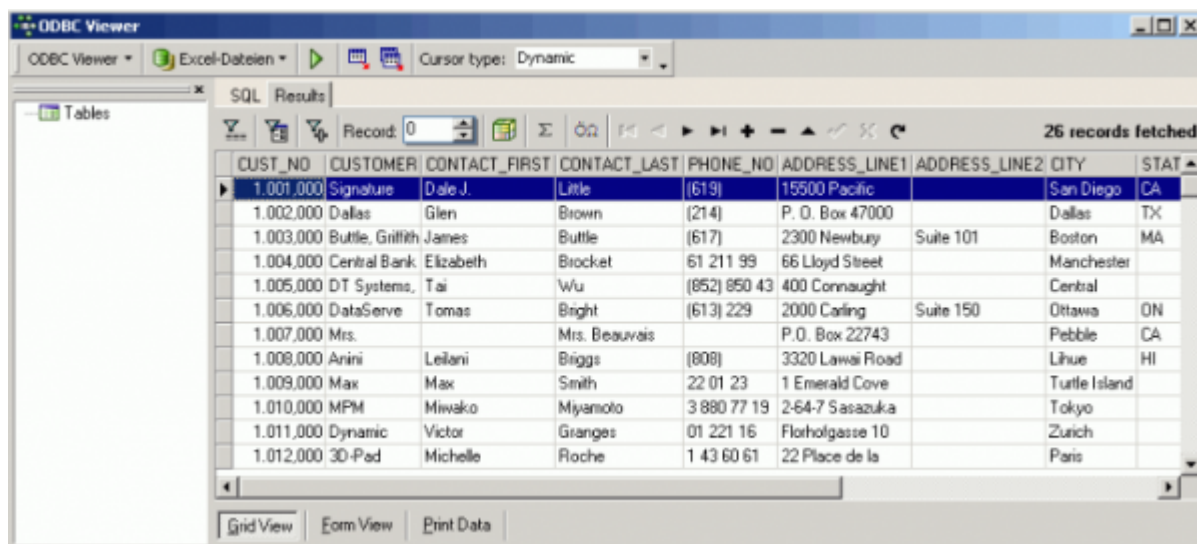
Alternatively an Excel file which is connected via ODBC can be viewed by typing the query:

```
select * from "sheet1$"
```

where sheet1\$ is the name of the spread sheet (visible on the tab at the bottom of the sheet). To view our example above:



The first line is used always used for the column names.



Extracting metadata

Metadata includes the definition of the [database](#) and [database objects](#) such as [domains](#), [generators](#), [tables](#), [constraints](#), [indices](#), [views](#), [triggers](#), [stored procedures](#), [user-defined functions \(UDFs\)](#), [blob filters](#). Metadata is stored in [system tables](#), which are themselves part of every Firebird/InterBase® database.

The Extract Metadata module can be used to generate a partial or full database [metadata](#) script, with or without [table](#) data, and including privileges and object descriptions if wished. It allows the user to extract metadata to file or clipboard. It is even possible to extract [blob](#) data and [array](#) fields' data (as blob data into a LOB file). This can then be used to install a customized empty (or full) database onto another machine or platform, by making a few simple manual alterations to the script.

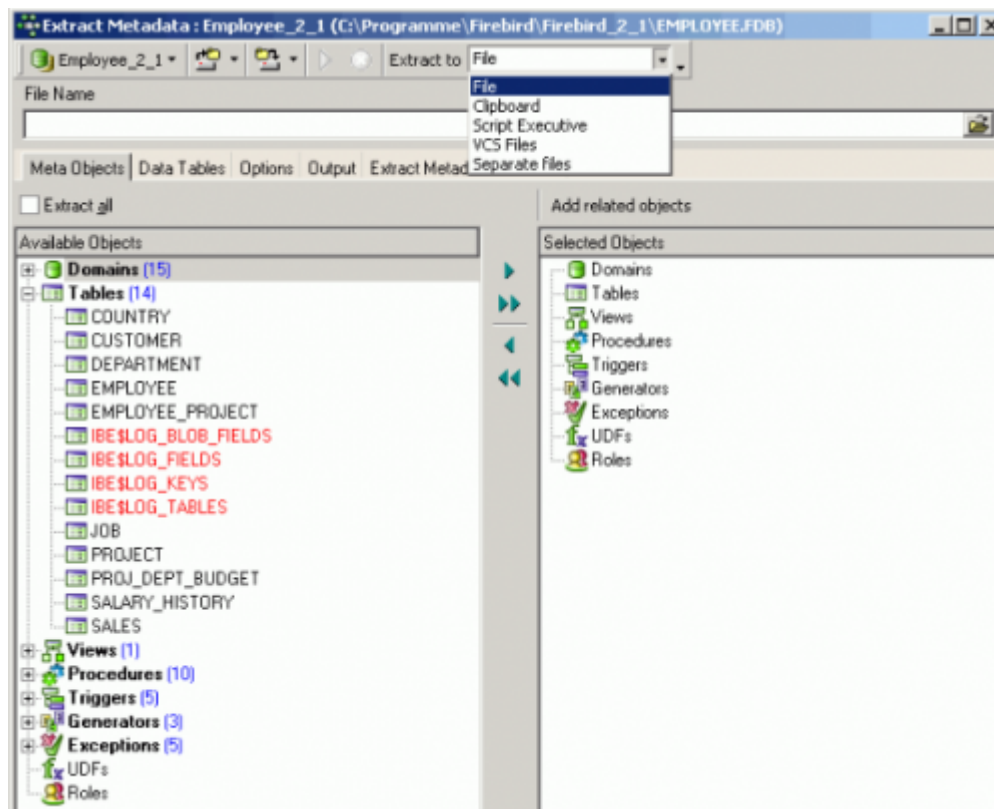


Table data can be extracted into separate files (TABLE_1.sql, TABLE_2.sql, TABLE_3.sql etc.) and support is also offered for secondary database file information; the corresponding ALTER DATABASE statements are included into the result script as comments.

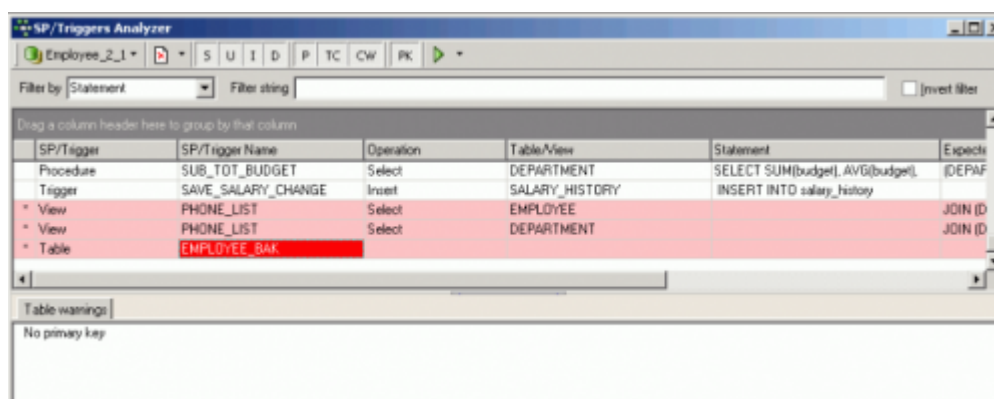
Full details of this tool can be found in the [IBExpert documentation](#) chapter, [Extract metadata](#).

Transferring data to a new table or another database using SQL

When you create a new [table](#) in IBExpert using the insert into command, for example:

```
insert into employee_bak
select * from employee
```

a table is created with the same structure as the original, with [field](#) names and data etc., but not the [primary](#) and [Foreign Keys](#), [constraints](#) etc. etc. When you are creating a table based on this command and you ignore this, you will see the table warning in the [SP/Triggers/Views Analyzer](#):



For a more detailed explanation of this feature, please refer to the [IBExpert documentation](#) chapter, [Creating a table from query results](#).

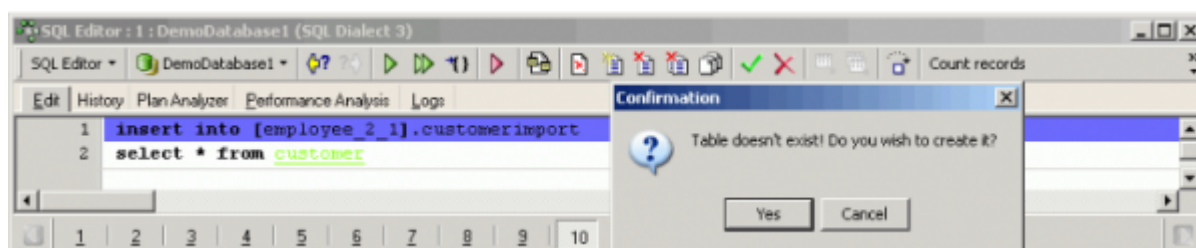
To transfer data from one registered database to another, open the SQL Editor for the db1 database, and

```
select * from customer
```

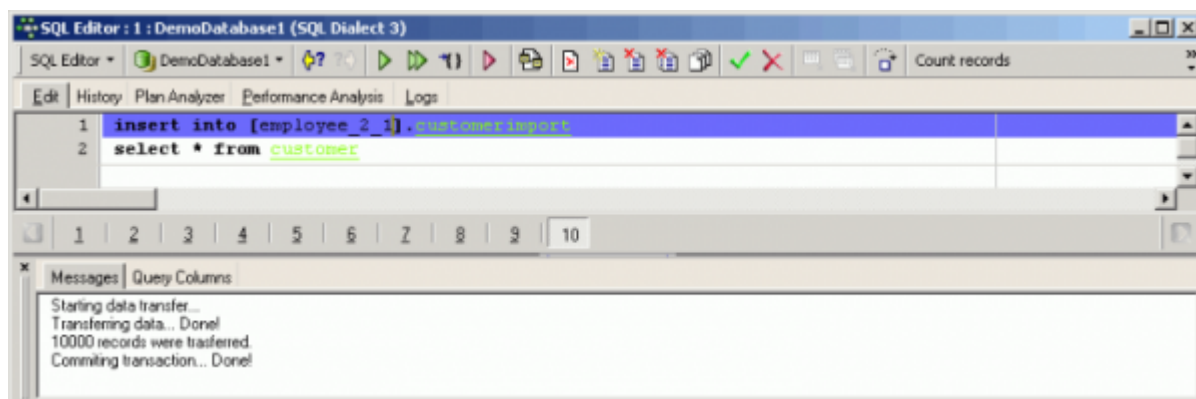
To transfer this data to another database, employee.fdb (with the IBExpert alias name, EMPLOYEE_2_1) simply write the following SQL when connected to the source (db1) database:

```
insert into [employee_2_1].customerimport  
select * from customer
```

If the table doesn't already exist, IBExpert will ask if you wish to create it:



and you have copied 10,000 customer records into another database without having to worry about first creating a new table of the same structure or performing some complicated export routine!



For a more detailed explanation of this feature, please refer to the IBExpert documentation, [Moving data between databases](#).

Creating a table from query results

As everyone knows it is possible to insert [data](#) into any [table](#) by executing the [INSERT](#) statement:

```
INSERT INTO TARGET_TABLE  
SELECT FIELD_1, FIELD_2 FROM SOURCE_TABLE  
WHERE SOMETHING_FIELD <> 5
```

However this will only work if the table TARGET_TABLE already exists in the database. IBExpert

enables execution of this kind of statement even if the TARGET_TABLE does not exist in the database. First IBExpert notifies the user that TARGET_TABLE doesn't exist in the [database](#) and offers to create this table using [query](#) structure. If confirmed, IBExpert creates the TARGET_TABLE and then populates it with data from the [SELECT](#).

A small example illustrates how this works, based on a SOURCE_TABLE with the following structure:

```
CREATE TABLE SOURCE_TABLE (  
    ID INTEGER,  
    SOME_TEXT VARCHAR(50),  
    SOME_PRICE NUMERIC(15,4),  
    SOME_DATE DATE);
```

When the following [statement](#) is executed:

```
INSERT INTO TARGET_TABLE  
SELECT * FROM SOURCE_TABLE
```

and there is no TARGET_TABLE in the database, IBExpert will create TARGET_TABLE as:

```
CREATE TABLE TARGET_TABLE (  
    ID INTEGER,  
    SOME_TEXT VARCHAR(50),  
    SOME_PRICE NUMERIC(15,4),  
    SOME_DATE DATE);
```

and after that inserts into this table records retrieved with the SELECT part.

Of course, it is possible to write different INSERT statements. For example:

```
INSERT INTO [TARGET_DATABASE].TARGET_TABLE  
SELECT ID, SOME_DATE FROM TEST_TABLE
```

In this case IBExpert will create table TARGET_TABLE as

```
CREATE TABLE TARGET_TABLE (  
    ID INTEGER,  
    SOME_DATE DATE);
```

Moving data between databases

IBExpert allows you to move [data](#) from one [database](#) to another by executing a special [statement](#) in the [SQL Editor](#).

Syntax

```
INSERT INTO <database_alias>.<table_name>  
[(<columns_list>)]
```

<select_statement>

Argument	Description
database_alias	Alias of a Register Database . This must be enclosed in Square Brackets . This argument is case-insensitive so aliases My alias and MY ALIAS are equivalent.
table_name	Name of the Table to be populated with data.
columns_list	List of columns in target table. This argument is not obligatory.
select_statement	Any SELECT statement.

Examples

The following statement moves data from SOURCE_TABLE of the current database into TARGET_TABLE of the database with the alias My test DB:

```
INSERT INTO [My test DB].TARGET_TABLE
SELECT * FROM SOURCE_TABLE
```

If the table TARGET_TABLE doesn't exist in the target database, IBExpert will create it after your confirmation with the structure of the SOURCE_TABLE.

Creating multiple scripts from a CSV file

The following is an example illustrating the creation of multiple csv files from a script using the [IBExpert Tools menu](#) item, [Script Executive](#):

```
shell del C:\list.dat nowait;      --deleting the old file
shell del C:\*.csv nowait;        --deleting the old csv files

connect 'localhost:C:\employee.fdb' user 'SYSDBA' password 'masterke';
--connect to employee example database
output 'C:\list.dat';             --record the following result as a simple text
file, based on each unique employee,
we create a new output ...;select ... ;output; line in the dat file

SELECT distinct
'OUTPUT C:\'||EMPLOYEE.last_name||'.csv delimiter ';'';'||
'SELECT distinct EMPLOYEE.last_name, customer.customer, customer.phone_no
' ||
'FROM SALES INNER JOIN CUSTOMER ON (SALES.CUST_NO = CUSTOMER.CUST_NO) ' ||
'INNER JOIN EMPLOYEE ON (SALES.SALES_REP = EMPLOYEE.EMP_NO) where
EMPLOYEE.last_name=XXXXXX||EMPLOYEE.last_name|''';'||
'OUTPUT;'
FROM SALES INNER JOIN CUSTOMER ON (SALES.CUST_NO = CUSTOMER.CUST_NO) INNER
JOIN EMPLOYEE ON

(SALES.SALES_REP = EMPLOYEE.EMP_NO);

output;      --close the dat file
```



```
input 'C:\list.dat'; --execute them
```

The data file is created automatically.

The outer query gets one record for each employee, in the inner select, all phone numbers for the employees if customers are selected. You can even use the IBExpert Script Executive to execute multiple scripts from a single script. Simply use the following syntax:

```
connect 'server:c:\my_db.gdb' ...;

input 'c:\my_scripts\f2.sql';
input 'c:\my_scripts\f1.sql';
input 'c:\my_scripts\f3.sql';
```

Please also refer to the IBEBlock example [Importing data from a CSV file](#) below.

Using IBEBlock, IBExpert's scripting language

IBEBlock is IBExpert's own scripting language. It contains a set of [DDL](#), [DML](#) and other statements that are executed on the server and on the client side, and which include some specific constructions applicable only in IBExpert or [IBEScript](#) (excluding the free versions of these products), independent of the database server version. The syntax of IBEBLOCK is similar to that of [stored procedures](#) but there are many important extensions.

IBEBlock is not just intended as an aid for data import and export, although it does offer a wide range of options covering this topic. With EXECUTE IBEBLOCK you will be able to work with different connections within a single IBEBLOCK at the same time, move (copy) [data](#) from one [database](#) to another, join tables from different databases, compare data from different databases and synchronize them, populate a [table](#) with test data using random values or values from other tables or even from other databases, access external databases via ODBC drivers, transaction control across multiple databases and so on and so forth.

You can execute single IBEBLOCKS in the [SQL Editor](#). You can debug them in the SQL Editor too. They are [debugged](#) in the same way as [stored procedures](#) and [triggers](#). Also you can include IBEBLOCKS into your scripts and execute these scripts as usual - using the [Script Executive](#) or [IBEScript.exe](#). IBEBlocks can be managed in the IBExpert [DB Explorer](#) (Script/Blocks page and Script node) and stored in the [IBExpert User Database](#). We strongly recommend using the [IBExpert User Database](#) as the main storage for IBExpert for security reasons (all your IBEBlocks are then stored in a Firebird database).

The online [IBEBlock documentation](#) covers the following topics:

- [Procedural extensions of IBEBlock](#)
- [IBEBlock functions](#)
- [Examples of usage of IBEBlock](#)

Examples illustrating the various importing and exporting functions include the following:

- [Importing data from a CSV file](#)
- [Importing data from a file](#)

- [ibec_CopyFile](#)
- [ibec_ImportData@](#)
- [EXPORT AS ... INTO](#)
- [Export data into DBF](#)
- [Export CSV data 1](#)
- [Export CSV data 2](#)

Using IBEScript.exe

IBEScript.exe can be used to execute any valid IBExpert script in batch files from the command line. For example a scheduled import or export job can be started without the need of any user input. IBEScript.exe can also be used to encrypt script files, so that they are unreadable for the user, but executable together with IBEScript.exe. If the end user does not have the necessary password(s), he cannot decrypt any files, ensuring security of sensitive information.

The [IBEScript.exe Distribution Software?](#) gives you the unlimited right to integrate IBEScript.exe in your application, as well as the IBExpert Server Tools. You can use it for executing IBEBlock scripts that connect to different databases at the same time, integrate any ODBC connection as source or destination, simple data import and export, metadata comparison and much more.

All functionalities of the IBEBlock scripting language as well as the [HK-Software Services Control Center](#) tools are also available on registered Server Tool computers, for example, metadata and data comparison, multiple database access, etc.

Just a few examples of what you can do with the IBEScript distribution software:

1. [IBEBlock technology to create procedures with access to data in different Firebird/InterBase® databases](#)
2. [ODBC access to all ODBC data sources for importing or exporting data from a script](#)
3. [Comparing databases from scripts](#)
4. [Create automatic reports](#)
5. [File import into blob fields from SQL scripts](#)

Using IBEScript.dll

The IBEScript.dll can be used in your applications to execute scripts from file or from a string buffer. There is a small demo application illustrating its use in the IBExpert installation's IBEScriptDll folder. Please

For regulations regarding distribution of this module together with your application, please refer to the IBExpert [products, services & prices](#) website.

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

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-05-database-technology:database-technology-articles:ibexpert-firebird-interbase:import-and-export-using-ibexpert2>

Last update: 2023/06/09 21:22

