IBEScript.exe

IBEScript.exe can be found in the IBExpert root directory, and needs to be started from DOS. (This feature is unfortunately not included in the free IBExpert Personal Edition.)

For regulations regarding distribution of any of the IBExpert Server Tools modules (hkSCC.exe, ibescript.exe, ibescript.dll and fbcrypt) together with your application, please refer to the IBExpert Server Tools documentation contents page.

Syntax

IBEScript script_filename [options]

- **S** = silent mode
- **V<Verbose_file>** = verbose output file. If <Verbose_file> exists, IBEScript will overwrite it.
- **v<verbose_file>** = verbose output file. If <verbose_file> exists, IBEScript will append message to this file.
- **E** = display only error messages.
- **N** = continue after error.
- **A<seconds>** = advanced progress: <seconds> specifies the time interval in seconds to display progress messages. If <seconds> is not specified, advanced progress information will be displayed every second. The following information is available:
- * **SC** - statements count, number of executed statements;
- * **POC** - percentage of completion;
- * **TS** - time spent;
- * **ETL** - estimated time left.

For INPUT scripts the name of the input script file name will also be displayed. Advanced progress information is displayed even if the -S option (silent mode) is specified.

- **T** = write timestamp into log.
- **D** = connections string (use it if your script does not contain CONNECT or CREATE DATABASE statements).
- **P** = connection password (use only with -D option).
- **R** = connection role (use only with -D option).
- **U** = connection user name (use only with -D option).
- **C** = character set (use only with -D option).
- **l** = client library file (gds32.dll if not specified).
- **L<1|2|3>** = SQL dialect (use only with -D option; 1 if not specified).
- **i** = idle priority.
- **G<variable_name>=<value>** = specify global variable values.
- **I<file_path_and_name>** = By default (without the -I option specified). IBEScript.exe processes IBEScript.ini files immediately after starting in the following order:

1. IBEScript.ini in the IBEScript.exe directory, if it exists,

2. IBEScript.ini in the current directory, if it exists. If only -I is specified without a file name, any INI-file will be ignored. If a file name is specified after -I (e.g. -I“C:\my files\myibescript.ini”) ONLY this file will

be processed if it exists. Parameters specified in the command line will overwrite corresponding ones from an INI file.

- **A<seconds>** = display advanced progress information every <seconds> second(s) (default 1)

- **e** = encrypt a script file (no execution will be performed)
- **d** = decrypt an encrypted script file (no execution will be performed)
- **p<password>** = encrypt/decrypt password
- **o<file_name>** = output file for encrypted and decrypted scripts

WARNING! All options are case-sensitive!

Example 1: IBEScript C:\My Scripts\CreateDB.sql **Example 2:** IBEScript C:\MyScripts\CreateDB.sql -S -VScriptLog.txt

The following features are also available: a login dialog will appear when no password and/or user name are specified in the CONNECT or CREATE DATABASE statements. It is also possible to change the connection character set (SET NAMES) and garbage collection option (SET GARBAGE_COLLECT) before the RECONNECT statement. Any SET commands mentioned which are followed by a RECONNECT statement will affect the new connection.

It is also possible to use environment variables in INPUT, OUTPUT and SET BLOBFILE statements (see example below).

UTF8 BOM is skipped when executing a script from file.

The OUTPUT command also supports the OctetsAsHex option, which allows the extraction of CHAR(n) CHARACTER SET OCTETS values in hexadecimal format.

IBEScript Examples

1. IBEBlock technology to create procedures with access to data in different Firebird/InterBase® databases

A simple script to copy data from one Firebird/InterBase® database to another:

```
 execute ibeblock
 as
 begin
   FBSrc  =
ibec_CreateConnection(__ctFirebird,'DBName="localhost:C:\DB1.FDB";
   ClientLib=C:\Program Files\Firebird\Bin\fbclient.dll;
   user=SYSDBA; password=masterkey; names=WIN1252; sqldialect=3');
   FBDest =
ibec_CreateConnection(__ctFirebird,'DBName="localhost:C:\DB2.FDB";
   ClientLib=C:\Program Files\Firebird\Bin\fbclient.dll;
   user=SYSDBA; password=masterkey; names=WIN1252; sqldialect=3');
   ibec_UseConnection(FbSrc);
```

```
   for select CustNo, Company, Addr1 from customer order by company into
:CustNo, :Company,
     :Addr1
   do
   begin
      use FBDest;
      INSERT INTO CUSTOMER (CustNo, Company, Addr1) VALUES (:CustNo,
:Company, :Addr1);
      use FBSrc;
   end
   use FBDest;
   COMMIT;
   ibec_CloseConnection(FBSrc);
   ibec_CloseConnection(FBDest);
 end
```

---

2. ODBC access to all ODBC data sources for importing or exporting data from a script

The same can also be done with any ODBC data source as the source and/or destination (this functionality has been tested with IBM DB2®, Oracle®, MS Access®, Sybase® etc.):

```
 execute ibeblock
 as
 begin
   OdbcCon = ibec_CreateConnection(__ctODBC, 'DBQ=C:\demo.mdb;
DRIVER=Microsoft Access
   Driver (*.mdb)');
   FBCon = ibec_CreateConnection(__ctFirebird,'DBName="AVX-
MAIN:D:\FB2_DATA\IBEHELP.FBA";
   ClientLib=C:\Program Files\Firebird\Bin\fbclient.dll;
   user=SYSDBA; password=masterkey; names=WIN1251; sqldialect=3');
   ibec_UseConnection(OdbcCon);
   for select CustNo, Company, Addr1 from customer order by company into
:CustNo, :Company,
   :Addr1
   do
   begin
      use FBCon;
   INSERT INTO CUSTOMER (CustNo, Company, Addr1) VALUES (:CustNo, :Company,
:Addr1);
      use OdbcCon;
   end
   use FBCon;
   COMMIT;
   ibec_CloseConnection(OdbcCon);
   ibec_CloseConnection(FBCon);
 end
```

3. Comparing databases from scripts

The following script compares the structure of two Firebird/InterBase® databases and stores a script that can be used to synchronize the database structure in the destination database: Save the following text as c:\comp.sql:

```
execute ibeblock
as
begin
  create connection ReferenceDB dbname 'localhost:c:\RefDB.fdb'
  password 'masterkey' user 'SYSDBA'
  clientlib 'C:\Program Files\Firebird\bin\fbclient.dll';

  create connection CustomerDB dbname 'localhost:c:\customerDB.fdb'
  password 'masterkey' user 'SYSDBA'
  clientlib 'C:\Program Files\Firebird\bin\fbclient.dll';

  cbb = 'execute ibeblock (LogMessage variant)
         as
         begin
           ibec_progress(LogMessage);
         end';

  ibec_CompareMetadata(ReferenceDB, CustomerDB, 'C:\CompRes.sql',
'OmitDescriptions; OmitGrants', cbb);

  close connection ReferenceDB;
  close connection CustomerDB;
end
```

Now run the following command line to create the script and synchronize the databases:

```
ibescript.exe c:\comp.sql
ibescript.exe c:\compres.sql
```

4. Create automatic reports

ibec_CreateReport prepares a report from a specified source and returns prepared report data. For preparing the initial report please refer to the IBExpert Report Manager.

This feature can be used for executing reports created with the IBExpert Report Manager in command-line mode, for example with batch files. The monthly sales report, invoices or other such reports can be designed in the Report Manager and executed with simple SQL statements. The result can then be saved in the database as a PDF file or other formats and sent by email or exported using ibec_ExportReport.

```
execute ibeblock
 as
 begin
   Params['HeaderMemo'] = '';
   Params['MEMO2'] = 2;

   SELECT IBE$REPORT_SOURCE FROM ibe$reports
   where ibe$report_id = 4
   into :RepSrc;

   Report = ibec_CreateReport(RepSrc, Params, null);
   ibec_SaveToFile('D:\reptest.fp3', Report, 0);
           Res = ibec_ExportReport(Report, 'D:\reptest.pdf', __erPDF,
'EmbeddedFonts=TRUE');
   Res = ibec_ExportReport(Report, 'D:\reptest.jpg', __erJPEG, 'CropImages;
Quality=90');
 end
```

---

5. File import into blob fields from SQL scripts

The following script imports the data from the files into the table TEST:

```
SET BLOBFILE 'C:\f1.jpg';
INSERT INTO TEST(ID,BLOBCOL) VALUES (1, :h00000000_7FFFFFFF);
SET BLOBFILE 'C:\f2.jpg';
INSERT INTO TEST(ID,BLOBCOL) VALUES (2, :h00000000_7FFFFFFF);
SET BLOBFILE 'C:\f3.jpg';
INSERT INTO TEST(ID,BLOBCOL) VALUES (3, :h00000000_7FFFFFFF);

The same syntax can be used for updating blob data.
```

---

6. Using environment variables in INPUT, OUTPUT and SET BLOBFILE statements

```
execute ibeblock
as
begin
ibec_SetEnvironmentVariable('MyScriptDir', 'D:\Scripts\MyScripts');
ibec_SetEnvironmentVariable('MyDataDir', 'D:\Data');
ibec_SetEnvironmentVariable('MyBlobData', 'D:\Data\Blobs');
end;

SET BLOBFILE '\mytable.lob';

OUTPUT '\mytable.sql';
select * from mytable
asinsert;
COMMIT;
```

```
INPUT '\ProcessData.sql';
```

These are just a few examples of the many tasks you can do with IBEScript. The full syntax and parameter list for ibec_CompareMetadata can be found in the online documentation, along with a full list of all current IBEBlock commands.

Encryption & decryption

There are two possible ways to encrypt/decrypt scripts and to execute encrypted scripts:

1. Encrypting without the password. In this case there is no possibility to decrypt an encrypted script but it is possible to execute this script with IBEScript.
2. Encrypting with the password. In this case it possible to decrypt the script and execute it with IBExpert if the correct password is specified.

The following options control the encrypting and decrypting:

- **-e** = encrypts a script file and creates a file with the extension .esql if the output file is not specified (no execution will be performed).
- **-d** = decrypts an encrypted script file if it was encrypted with a password (no execution will be performed).
- **-p<password>** = encrypt/decrypt password.
- **-o<file_name>** = output file name for encrypted and decrypted scripts.

Again: all options are case-sensitive!

Example 1

```
IBEScript "C:\MyScripts\CreateDB.sql"
```

Example 2

```
IBEScript C:\MyScripts\CreateDB.sql -S -UScriptLog.txt
```

This product can be purchased as part of the distribution software listed above.

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

Permanent link:
https://www.ibexpert.com/docu/doku.php?id=01-documentation:01-02-server-tools:ibescript.exe

Last update: **2023/05/31 16:42**