



# Report Admin



*Reporting Server based on REPCODER.DLL and Firebird / InterBase SQL*

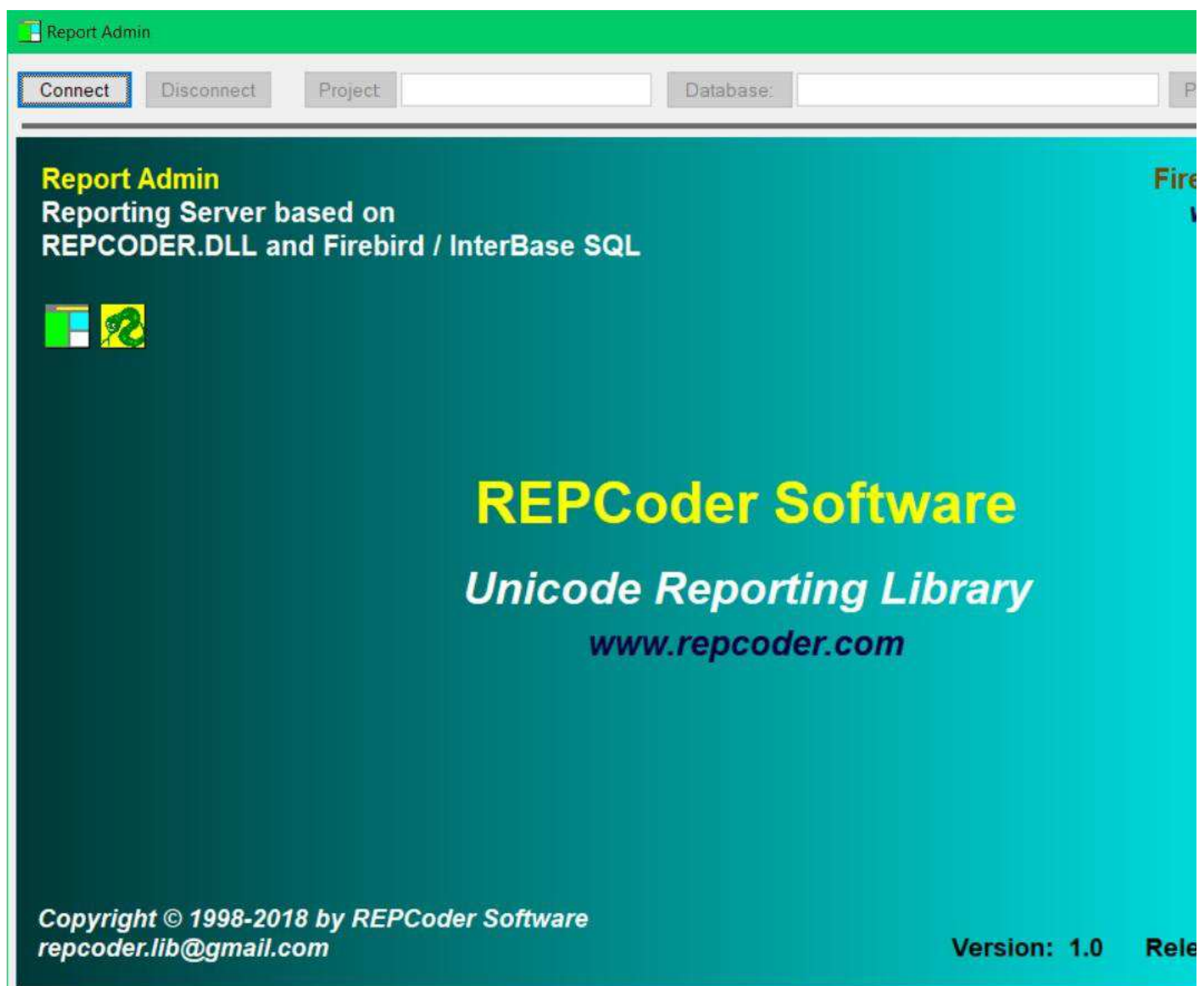
*Version: 1.0 (release date: 2018-05-19)*

## Start:

There is no installation procedure. You need only **REPADMIN.EXE** and **REPCODER.DLL**. In the 64-bit version: **REPADMIN64.EXE** and **REPCODER64.DLL**.

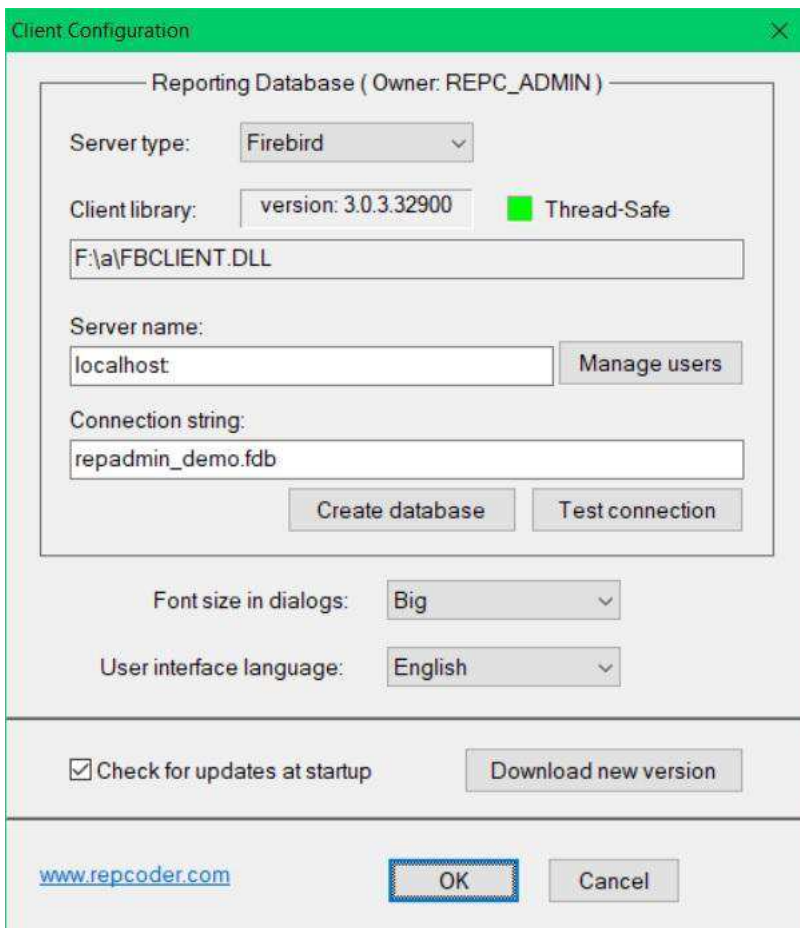
It is a database application created to manage reports designed with REPCoder. The designed report projects are normally stored in SFM files. REPADMIN.EXE stores them in a Firebird (or InterBase) database as **blobs**. So you can put your reports on a server and access them as remote database data. The owner of this **Reporting database** is **REPC\_ADMIN**. This username must be added to the security database of the Firebird (InterBase) server at the same beginning.

REPADMIN.EXE fetches a report from the **Reporting database**, saves it locally in a temporary SFM file and opens using [repc\\_open\\_report](#) function of REPCODER.DLL. Then the SFM file is deleted from the local disc (after the report is closed).



## Client configuration:

Here you first specify the **type** of the **Reporting database**. It can be based on **Firebird** or **InterBase** SQL server. The version and path of the client library is also shown. The latest (thread-safe) versions are recommended. The name of the client library is: **fbclient.dll** (Firebird) or **gds32.dll** (InterBase). For 64-bit version the InterBase client is **ibclient64.dll**.



Client Configuration

Reporting Database ( Owner: REPC\_ADMIN )

Server type: Firebird

Client library: version: 3.0.3.32900 ☒ Thread-Safe

F:\a\FBCLIENT.DLL

Server name: localhost Manage users

Connection string: repadmin\_demo.fdb

Create database Test connection

Font size in dialogs: Big

User interface language: English

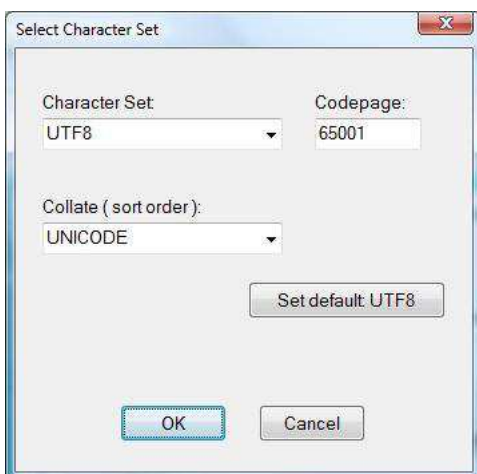
☒ Check for updates at startup Download new version

[www.repcoder.com](http://www.repcoder.com) OK Cancel

## Create a reporting database:

Then you specify the **Server name** and the **Connection string** to the **Reporting database**. You can also *"Manage users"* of the server here. It is good to add the username: **REPC\_ADMIN** at the same beginning, because it is the Owner of the reporting database. For this you have to know the password for the server superuser: **SYSDBA**. Then you use the *"Create database"* button to create the reporting database on the server. If the REPC\_ADMIN user was not created yet you will be asked to enter the SYSDBA password. The program will add REPC\_ADMIN to the security database. Its default password is "admin". You can later change this password in the *"Manage users"* dialog.

Before creating the database you are asked to specify its **Character Set** and **Collate**. The default recommended values are: **UTF8** and **UNICODE**. But you can choose it on your own. It is important for the TEXT data in the reporting database: names of reports, groups, projects, documents.



Select Character Set

Character Set: UTF8

Codepage: 65001

Collate ( sort order ): UNICODE

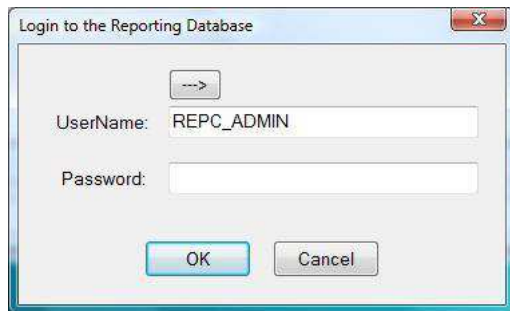
Set default: UTF8

OK Cancel

You can of course create as many reporting databases as you need, and on different servers. But it is not necessary. One database is enough for all needs, because you can create different **Projects** there to organize your reports.

## Login to the reporting database:

After the reporting database has been created, you must use the "Connect" button in the upper-left corner of the main window. First you should login as REPC\_ADMIN, because only this owner user can create the projects and design the reports. Other users have rather the read-only access to the reporting database. They can only execute the reports provided by REPC\_ADMIN. The only difference is that they can also create their own **documents** (report results) and grant access to them to other users.



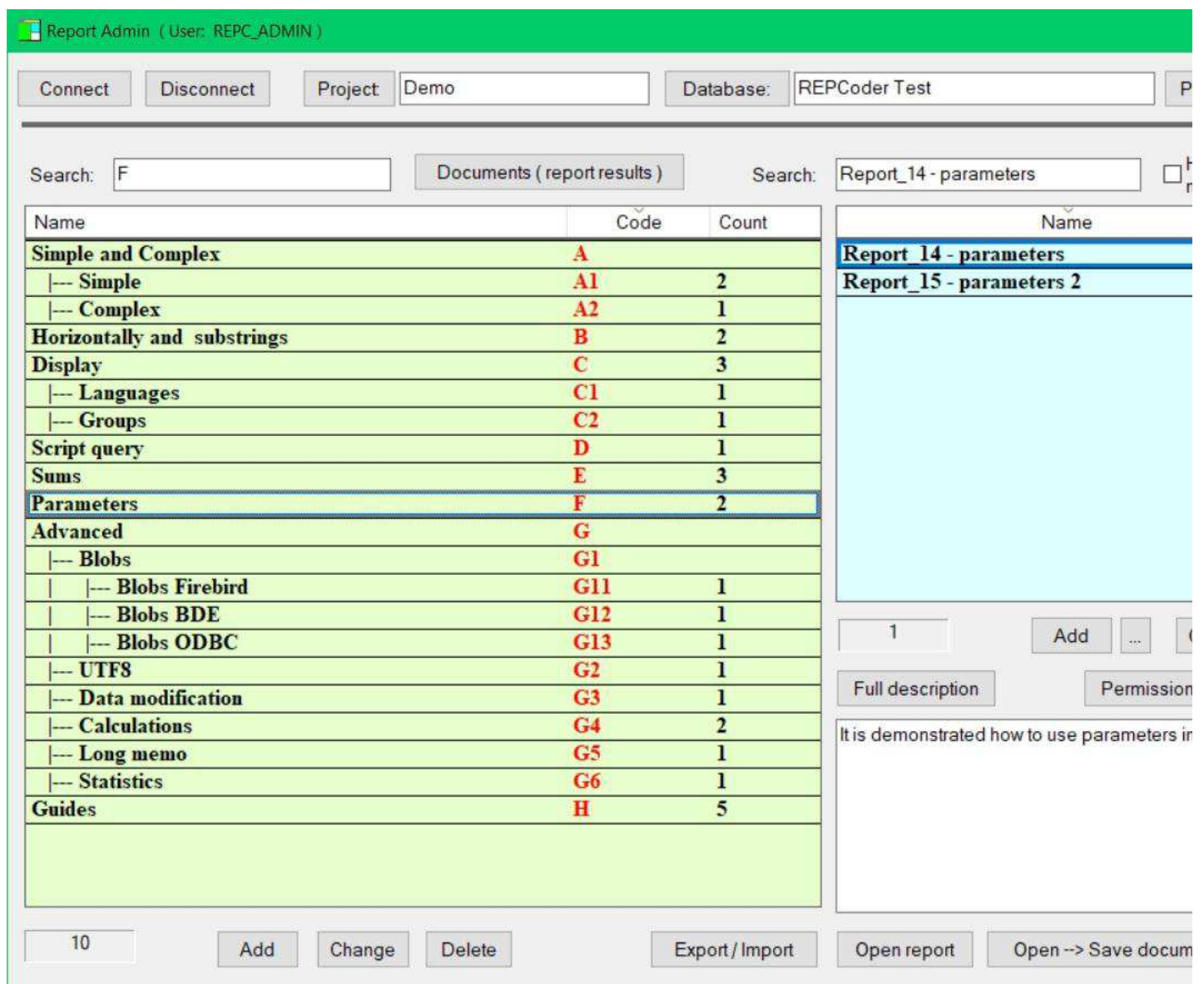
Login to the Reporting Database

UserName: REPC\_ADMIN

Password:

OK Cancel

After successfull login you see the following window. The predefined **Demo** project is opened there. For your own reporting database this project does not exist (the list of projects will be empty).



Report Admin (User: REPC\_ADMIN)

Connect Disconnect Project Demo Database: REPCoder Test

Search: F Documents ( report results ) Search: Report\_14 - parameters

Name	Code	Count
<b>Simple and Complex</b>	<b>A</b>	
--- Simple	A1	2
--- Complex	A2	1
<b>Horizontally and substrings</b>	<b>B</b>	2
<b>Display</b>	<b>C</b>	3
--- Languages	C1	1
--- Groups	C2	1
<b>Script query</b>	<b>D</b>	1
<b>Sums</b>	<b>E</b>	3
<b>Parameters</b>	<b>F</b>	2
<b>Advanced</b>	<b>G</b>	
--- Blobs	G1	
--- Blobs Firebird	G11	1
--- Blobs BDE	G12	1
--- Blobs ODBC	G13	1
--- UTF8	G2	1
--- Data modification	G3	1
--- Calculations	G4	2
--- Long memo	G5	1
--- Statistics	G6	1
<b>Guides</b>	<b>H</b>	5

Report\_14 - parameters  
Report\_15 - parameters 2

1 Add ...

Full description Permission

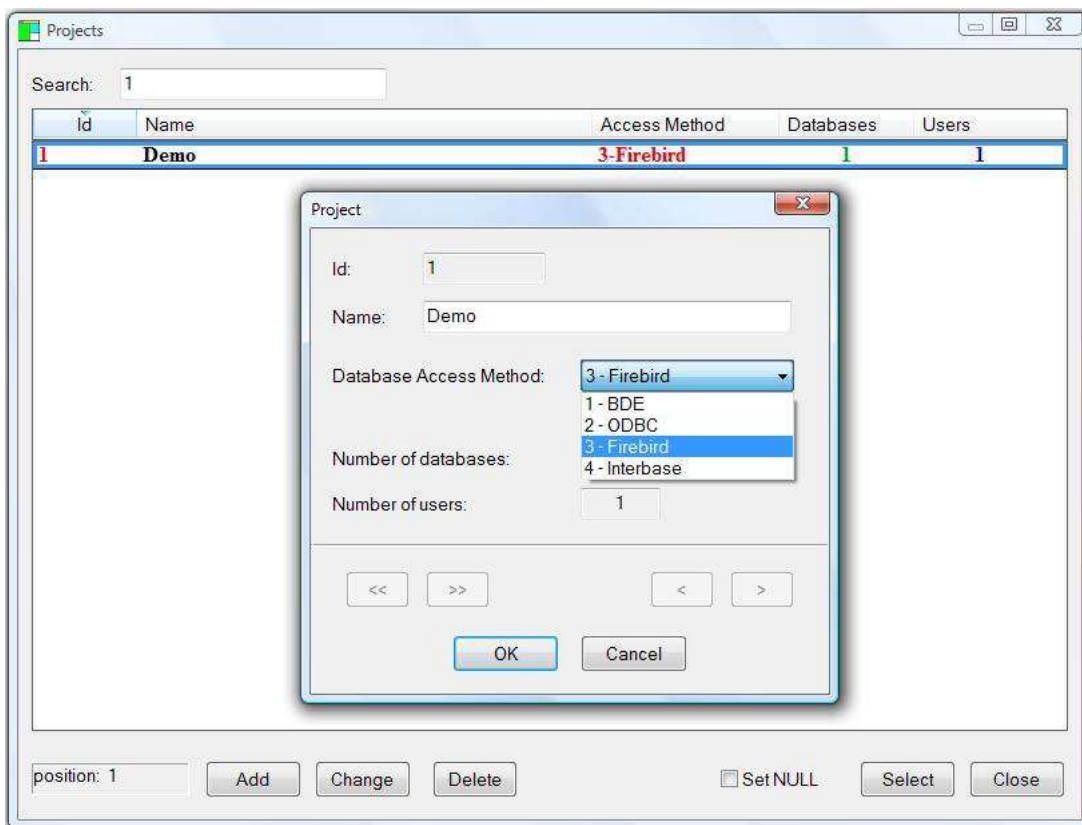
It is demonstrated how to use parameters in

10 Add Change Delete Export / Import Open report Open --> Save docum

## Projects:

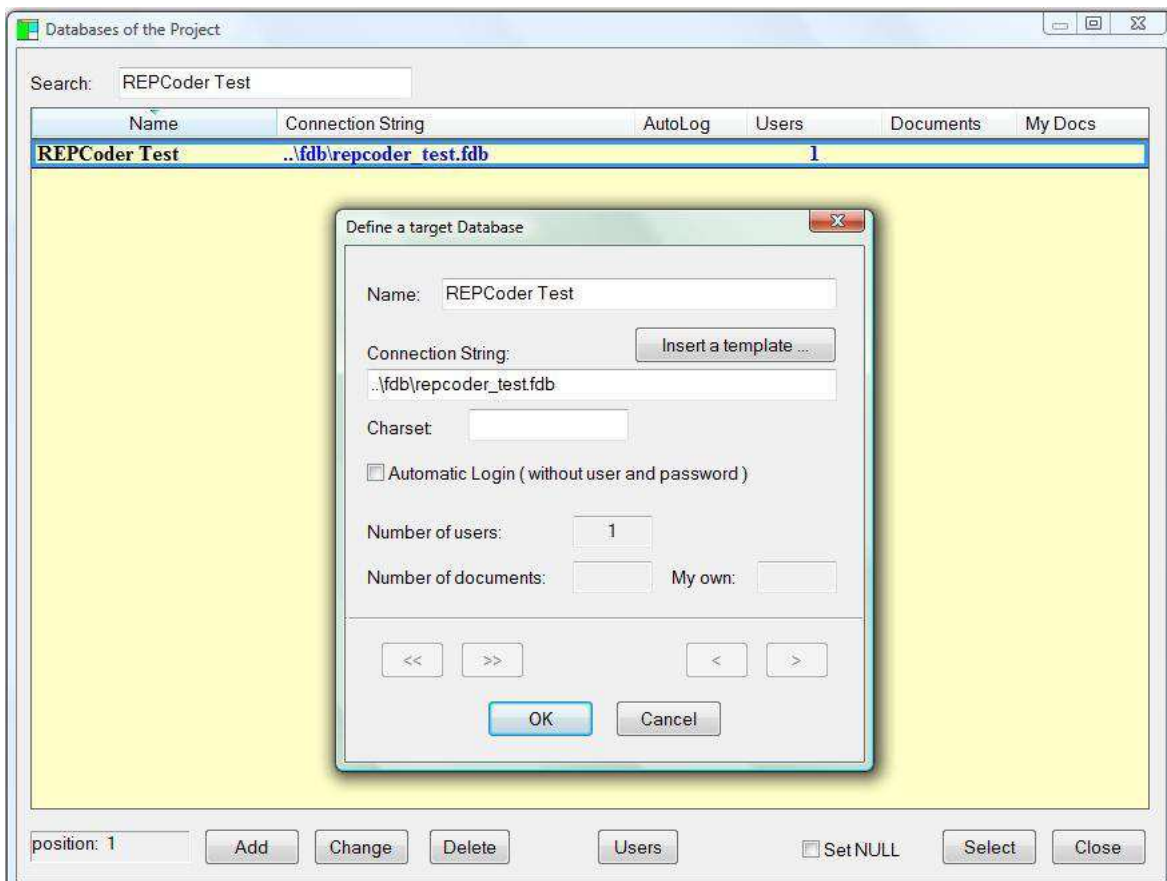
The main structure of the "Reporting database" is: **REPORTS --> GROUPS --> PROJECTS**.

So first you create a **Project**. Each project has its unique **Name** and the selected **Database Access Method**: 1-BDE, 2-ODBC, 3-Firebird, 4-InterBase.



For each project you define its **Groups** and at the end you add **reports** to the groups. Each group has its unique **Name** (in the Project) and a short unique **Code**. These codes define the **tree structure** of groups (for example: A,B, A1,A2,B1,B2, A11,A12,B11,B12 ...).

For each project you also define and configure the set of target **Databases**. All the target databases of the Project should have the same structure. They should differ only with the data stored inside. Usually they will be the databases of different business clients. Each target database has its unique **Name** in the Project. For each target database you create the **list of users** and configure their target **default logins** and (optional) target **roles**. The user names are those stored in the security database of the underlying Firebird (InterBase) reporting server.

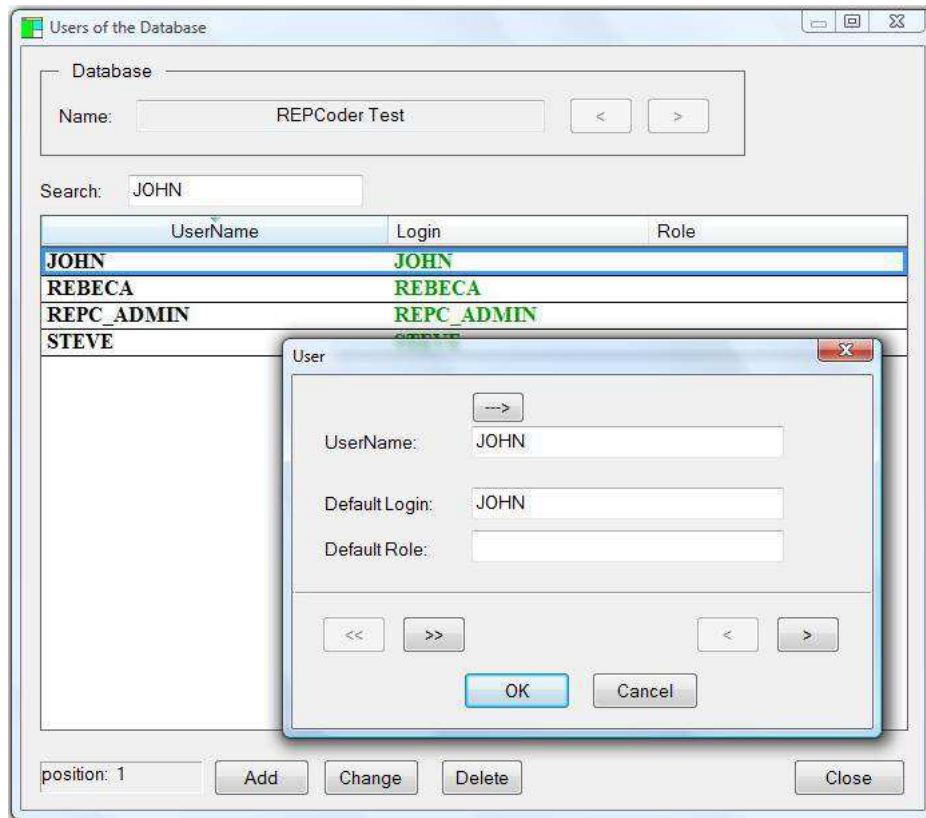


The most important feature of the target database is the **Connection String**. It will be passed to each report as its parameter number 1. So when you design a report with REPADMIN.EXE this parameter is already defined and should not be removed. It is used to define the connection string inside the reports. The reports are designed by REPC\_ADMIN with REPCoder (provided here by REPCODER.DLL).

You can also specify another optional connection parameter: **CharSet** if necessary. The charset is optional, because it can also be specified inside the reports.

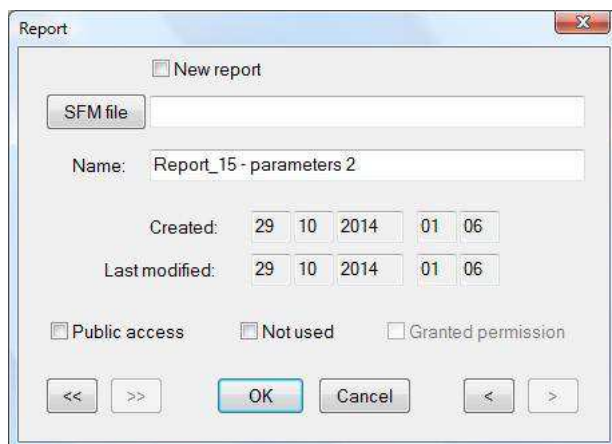
Another parameter here is the so-called **Automatic Login**. If checked, users will not be prompted for logins and passwords.

So the additional structure of the "Reporting database" is: **DBUSERS --> DATABASES --> PROJECTS**.



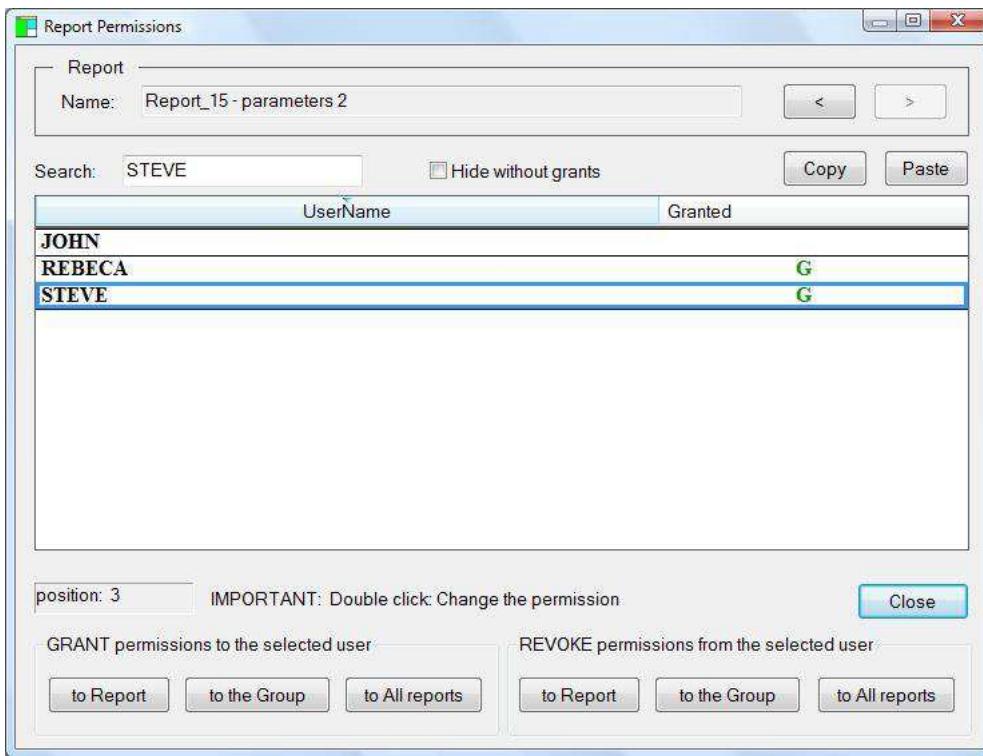
## Reports:

The reports can be designed only by the owner user: REPC\_ADMIN. Each report has its unique **Name** in the Project. Only the REPC\_ADMIN user can define projects, groups, databases, list of database users and their permissions. When creating a report it can be loaded from SFM file (designed earlier with REPCoder) or be designed here as a completely **New report**. You can also mark a report as having **Public access** or with a **Not used** flag. This flag is for old, obsolete reports. You can hide them later in the main window using the "Hide not-used" checkbox.



To execute a report, a user of the "Reporting database" must have a **Permission** granted by REPC\_ADMIN. Only **Public** reports can be executed by all users. The user must have also been added to the "list of users" of the target database for which the report is to be executed.





Report Permissions dialog box. The 'Report' section shows 'Name: Report\_15 - parameters 2'. The 'Search' section shows 'STEVE' and a 'Hide without grants' checkbox. Below is a table with columns 'UserName' and 'Granted'.

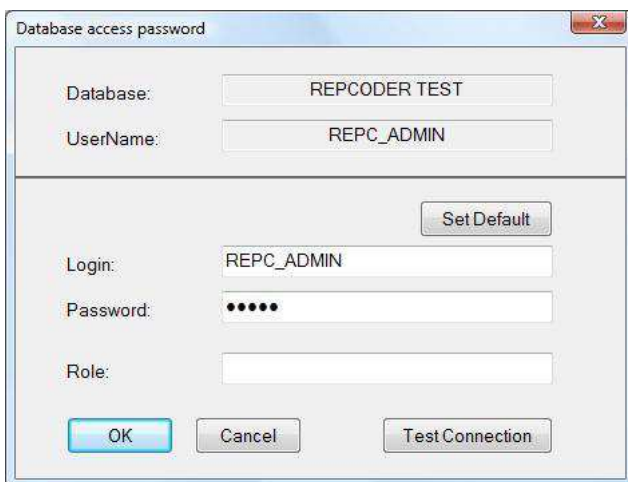
UserName	Granted
JOHN	
REBECA	G
STEVE	G

At the bottom, there is a 'position: 3' field, an 'IMPORTANT: Double click: Change the permission' message, and a 'Close' button. Below this are two sections: 'GRANT permissions to the selected user' and 'REVOKE permissions from the selected user', each with buttons for 'to Report', 'to the Group', and 'to All reports'.

## Target database access password:

When a user of the reporting database wants to execute any report he must first:

- select a Project
- select a target Database
- set a **Password** to the target Database



Database access password dialog box. It contains fields for 'Database:' (REPCODER TEST), 'UserName:' (REPC\_ADMIN), 'Login:' (REPC\_ADMIN), 'Password:' (masked with dots), and 'Role:'. There is a 'Set Default' button next to the 'Login' field. At the bottom are 'OK', 'Cancel', and 'Test Connection' buttons.

This password can be set using the "Password" button in the upper-right corner of the main window. If not set the user is prompted to do it before he tries to open the first report.

After the password is set the target database name is marked with the **yellow** color.

Report Admin ( User: REPC\_ADMIN )

Connect Disconnect Project Demo Database: REPCoder Test < SYSDBA >

Search: F Documents ( report results ) Search: Report\_14 - parameters

Name	Code	Count
<b>Simple and Complex</b>	<b>A</b>	
--- Simple	A1	2
--- Complex	A2	1
<b>Horizontally and substrings</b>	<b>B</b>	2
<b>Display</b>	<b>C</b>	3
--- Languages	C1	1
--- Groups	C2	1
<b>Script query</b>	<b>D</b>	1
<b>Sums</b>	<b>E</b>	3
<b>Parameters</b>	<b>F</b>	2
<b>Advanced</b>	<b>G</b>	
--- Blobs	G1	
--- Blobs Firebird	G11	1
--- Blobs BDE	G12	1
--- Blobs ODBC	G13	1
--- UTF8	G2	1
--- Data modification	G3	1
--- Calculations	G4	2
--- Long memo	G5	1
--- Statistics	G6	1
<b>Guides</b>	<b>H</b>	5

10 Add Change Delete Export / Import Open report Open --> Save docum

Report\_14 - parameters  
Report\_15 - parameters 2

1 Add ...

Full description Permission

It is demonstrated how to use parameters in

You should understand that this **Password** is then passed to REPCODER.DLL when any report is executed. So you don't need to define a password on the level of a report project. The same is with the **Login** and **Role** to the target database. These 3 parameters, together with the target database **Connection String** and **CharSet** are passed to each report called by REPCODER.DLL. This way a report "knows" the connection parameters for the target database. And you don't need to worry about it when you design a report.

## UserName and Login:

So you should see the difference between the two terms that are used in "Report Admin": **UserName** and **Login**. The "*UserName*" is only the name of a user of the "*Reporting database*" (where the report projects are stored). These names are stored in the security database of the underlying Firebird or InterBase SQL server. But the "*Login*" is the corresponding name of this user for a *target database*. So one user of the "reporting database" can have many different logins for different "target databases".

## Documents:

All users can execute the reports and optionally save their SFR results as the **Documents**. The report results are stored in temporary SFR files and then put into the "Reporting database" as blobs.

Documents

Search:  Year:  Month:  Created by:

Date	Name	Length	Created by	Public
2014-10-20	Report_25 - statistics	17080	REPC_ADMIN	
2014-10-30	Report_14 - parameters	8628	STEVE	P
2014-10-31	Report_25 - statistics3	16846	REPC_ADMIN	
2014-11-02	Report_25 - statistics6	17049	REPC_ADMIN	
2014-11-02	Report_25 - statistics8	17048	REPC_ADMIN	
2014-11-02	Report_25 - statistics9	17049	REPC_ADMIN	
2014-11-02	Report_25 - statistics4	16817	REPC_ADMIN	
2014-11-02	Report_25 - statistics5	17049	REPC_ADMIN	
2014-11-02	Report_25 - statistics7	17049	REPC_ADMIN	
2014-11-02	Report_17 - blobs	133967	REPC_ADMIN	
2014-11-02	Report_23 - data transfer	135352	STEVE	P
2014-11-02	Report_24 - long memo	15983	REPC_ADMIN	P

position: 1

The user who created a document can also grant permissions to it to other users of the "reporting database". He can also drop a document any time. This actions are also always allowed to REPC\_ADMIN (the owner of the reporting database).

The documents can also be extracted back from the "reporting database" to SFR files - "Save to files" button.

## Export / Import:

The reports designed in the program and stored in the "reporting database" can be easily extracted to SFM files.

Export / Import

Extract reports from

☐ All groups
 ☒ Selected groups

Save to SFM files

Export Project to file

Import Project from file

Also a whole Project (all groups and reports) can be saved to a binary file (.PRO) and then imported in another project, in the same or a different reporting database. The target databases of the project are not saved in this archive.

## A sample report:



Report Admin (User: REPC\_ADMIN)

Connect Disconnect Project Demo Database: REPCoder Test <SYSDBA>

Search: G11 Documents (report results) Search: Report\_17 - blobs

Name	Code	Count
Simple and Complex	A	
--- Simple	A1	2
--- Complex		
Horizontally and substrings		
Display		
--- Languages		
--- Groups		
Script query		
Sums		
Parameters		
Advanced		
--- Blobs		
--- Blobs Firebird		
--- Blobs BDE		
--- Blobs ODBC		
--- UTF8		
--- Data modification		
--- Calculations		
--- Long memo		
--- Statistics		
Guides		

13 Add Change

REPCoder: tmp\_Report\_17 - blobs.sfm (Page 1/6)

File Edit Form < Page 1/6 >

Page 1 / 6


## Report - blobs

How to display picture and MEMO database fields

Internal query: *select \* from BLOBS*

The result set has two blob (binary large object) columns here: *Notes*, *Graphic*. The first is a MEMO field, the second is a BMP picture data. The MEMO field can have different length in the rectangle where it belongs as *dynamic*. For that we check the option: "*Dynamic rectangle*" in the lower part of the "Texts" dialog. Dynamic rectangles are easily project form, because they have *blue triangles* in the lower corners. When the report is dynamically calculated depending on the amount of text. When the report is cut it is always on the same page, because its rectangles are elements of one group.

1).	Species No: 90020	Category: Triggerfish	picture (38878 b)
Name: Clown Triggerfish			
Species Name: Ballistoides conspicillum		cm: 50	inch: 19,69
notes (708 characters): Also known as the big spotted triggerfish. Inhabits outer reef areas and feeds upon crustaceans and mollusks by crushing them with powerful teeth. They are voracious eaters, and divers report seeing the clown triggerfish devour beds of pearl oysters. Do not eat this fish. According to an 1878 account, "the poisonous flesh acts primarily upon the nervous			



## IMPORTANT:

If a directory where REPADMIN.EXE is located has also the "sfm" subdirectory, then the temporary SFM files will be stored by REPADMIN.EXE just there. You can also put there some image files that your reports will be using. So it is recommended that this subdirectory exists.

## API for programmers:

The programmers using REPCODER.DLL can also easily access the reports stored in the reporting database (instead of standard SFM files) from their applications. To enable this, a new (and only one) exported function was added to REPCODER.DLL: [repc\\_admin\\_setup](#). Here you pass all the information necessary to access reports in the project of the "Reporting database". The return value is 1 (success) or 0 (error). The function header in C is:

```

BOOL repc_admin_setup(
    BOOL bInterbase, /* 0: Firebird-based reporting database, 1: InterBase-based reporting database */
    wchar_t* ConnString, /* reporting database: Connection String */
    wchar_t* User, /* reporting database: Username */
    wchar_t* Password, /* reporting database: Password */
    wchar_t* ProjectName, /* reporting database: Unique project name */
    wchar_t* DbName, /* target database: Unique database name in the project */
    wchar_t* DbLogin, /* target database (optional): Login for DbName (if not specified the defined value will be used) */
    wchar_t* DbPassword, /* target database: Password for DbName */
    wchar_t* DbRole, /* target database (optional): Role for DbName (if not specified the defined value will be used) */
    BOOL bVerify, /* (0: no verification, 1: verify if "ProjectName" and "DbName" exist and if "User" is on the list of "DbName" users) */

```

**BOOL bShowErrorMessage** /\* (0: quiet verification without messages, 1: MessageBox with error when the verification fails) \*/  
);

After you call this function with success you can easily access the reports stored in the project ("ProjectName") in the reporting database ("ConnString"). You normally use the functions of REPCODER.DLL like "repc\_open\_report" or "repc\_call\_report" in exactly the same way as you open SFM files with them. The only difference is that you do not provide the ".SFM" extension in the report name. It should be just the unique name of the report in the given "Project" in the reporting database.

This UNICODE function has also its ANSI version: "repc\_admin\_setup\_a" (as all other functions in REPCODER.DLL).

This function is also provided for C#, DELPHI, JAVA. For more info see the "Guide 4 - DLL.sfm" in the [repcoder.zip](#) package.

The function "repc\_admin\_setup" only sets some globals in the REPCODER.DLL. It does not leave the reporting database opened on output and does not allocate any resources. So there is no cleaning function provided to free resources at the end. The reporting database is opened each time a report is called to fetch its blob data (SFM file) and then it is closed immediately.

But REPADMIN.EXE behaves in the different way - it keeps the reporting database opened all the time it is running. If you are a programmer you need REPADMIN.EXE only at the beginning - to create the reporting database or later - to make changes to it. Because it is not possible to design reports in the reporting database using REPCODER.DLL from your application. You can only execute them this way.

---