Securing Interbase gbak 2.0

Introduction

When you want to use the Interbase gbak utility in a commandfile for batch processing you are stucked to the fact that you have to include the username and the password in clear text. If security is importend there are several ways to minimize the risks, like fileprotection, but they all rely on the fact that you - and only you - are the administrator with the highest rights.

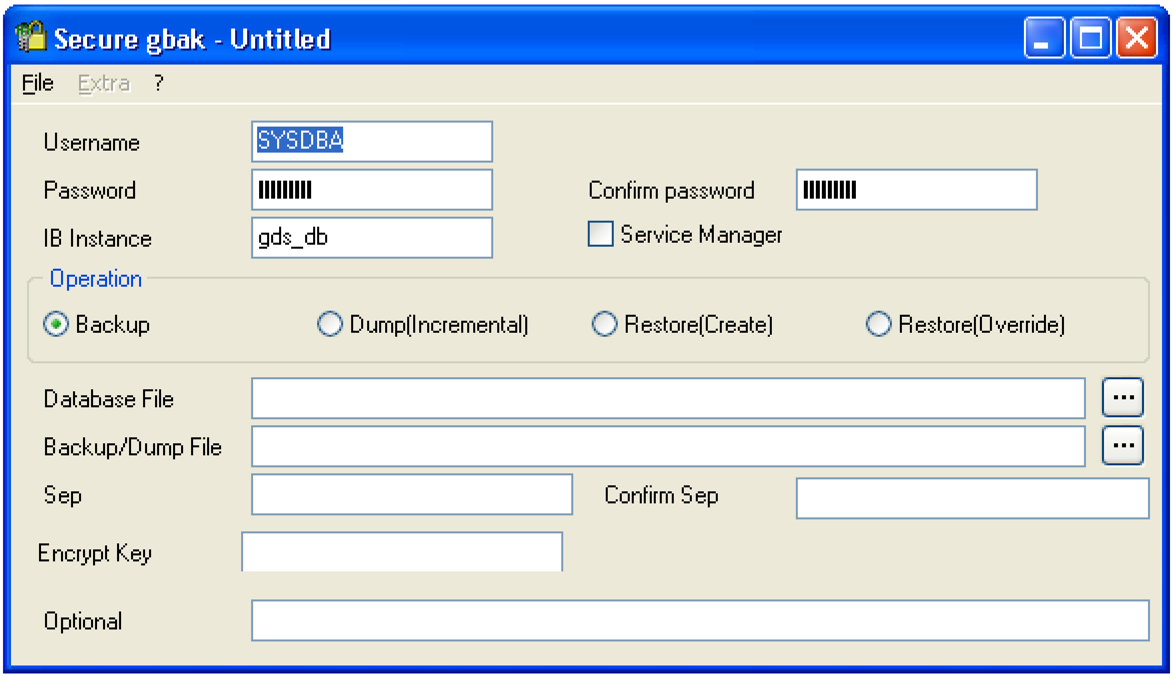
As long as Interbase does not support the usage of encrypted usernames and passwords, you will have to do it your self. The easiest way to do this, is writing a wrapper round gbak with the flexibility of gbak and the possibility of hiding or securing your username and password.

For this purpose I wrote **securegbak**. It easy to use and makes sure that no one else ought to know your username and password.

With the release of InterBase XE things got more complicated when you use things like Embedded User Authentication (EUA) or even encryption. Apert from that 64-bits systems are getting commenly used todays, so I decided to a update securegbak to implement all new features and release 32-bits and 64-bits versions. To distinguish between those the executables are now called sgb32.exe and sgb64.exe.
Securegbak 2.0 is backwards compatible with version 1.0.

How it works

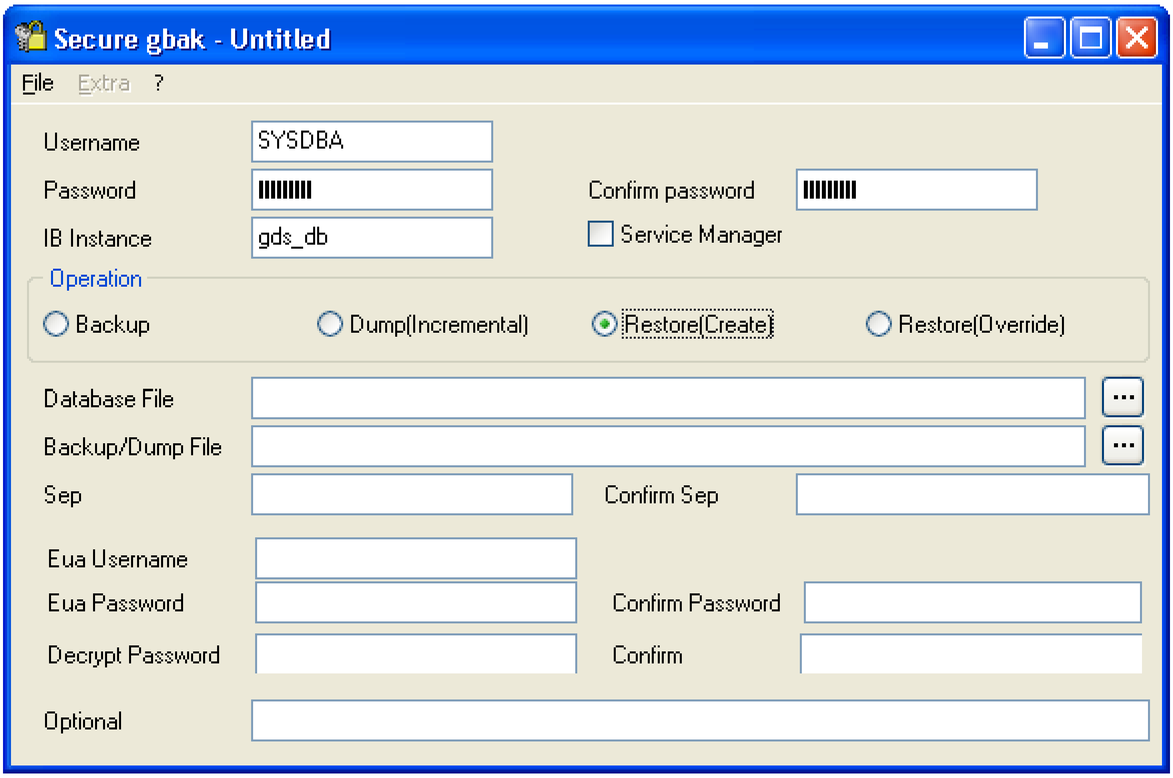
Securegbak uses a binairy configuration file. The usernames and passwords are saved in encrypted form in the configuration file. When securegbak is invoked without a parameter a GUI will be displayed to create or modify the configuration file. Initialy the form is set to “Backup”.



Username: SYSDBA
Password: ••••••••
Confirm password: ••••••••
IB Instance: gds_db
□ Service Manager
Operation: ● Backup ○ Dump(Incremental) ○ Restore(Create) ○ Restore(Override)
Database File:
Backup/Dump File:
Sep:
Confirm Sep:
Encrypt Key:
Optional:

There is no default configuration file. The program starts with default values for the username, the password and the instancename of InterBase. The instancename is only relevant if you have multiple instances of InterBase running on the machine, but it may not be blank. Leave it to “gds_db” if there is only one instance of InterBase.

When you select one of the restore options in the “Operation” groupbox the form will be enhanced to specify the parameters that might be needed for the restore operation.



Username SYSDBA

Password IIIIIIIII

Confirm password IIIIIIIII

IB Instance gds_db

☐ Service Manager

Operation

○ Backup ○ Dump(Incremental) ● Restore(Create) ○ Restore(Override)

Database File

Backup/Dump File

Sep

Confirm Sep

Eua Username

Eua Password

Confirm Password

Decrypt Password

Confirm

Optional

In version 1.0 you had to specify the full commandline for gbak without the username and password. In version 2.0 you just fill in the form and securegbak does the rest. In the “Optional” textbox you can specify any additional parameters to pass to the commandline.

You can give the configuration file any name you want (the extension will always be ".sgb"). This gives you the oppertunity to have several configuration files in the same folder for several jobs.

To execute a job just specify the full file specification of the configuration file to securegbak. If the full file specification includes one of more spaces then make sure you use double quotes. If you want to redirect output to a logfile make sure there is a space between the file specification and the ">". Example:
sgb32 "c:\my spaced path\myconfig.sgb" >myerror.log.

Version 2.0 still can execute configuration files created with version 1.0, but when you want to make changes to an old configuration file you must make it compatible with version 2.0. The commandline from version 1.0 will be displayed in the “Optional” textbox so you can cut and paste the names of the databasefile and backupfile to the approperiate textboxes.

Once the configuration file is saved the menu option “Extra” becomes available giving the oppertunity to display the commandline that will be executed and even to test it. Be aware that testing means actually executing the command so be careful when testing a restore operation.
When you test the commandline the program will be suspended until the execution of the commandline finished. It does not return usefull information. The messages only indicate if the commandline was executed by the Operating System or not. When the operation failed the first part of the commandline might be incorrect. In that case use the option “Show commandline” and check if the specification of the location of gbak.exe is correct.
If the operation “Completed” you have to check yourself if it was succesfull.

The use of securegbak is on your own risk!

Gelein van de Voorde