# Setup and usage

Start the HK-Software Services Control Center, found in the IBExpert Services menu, and select *IBExpert Backup Restore* in the *HK services list*.

We now need to configure the default task settings. We know that some parameters will remain the same for all further tasks (for example: Firebird Install folder, path to gbak.exe, SMTP settings, etc.), so we should configure those first.

Expand the *DefaultTaskSettings* item on the *Service setup* and *control page*.
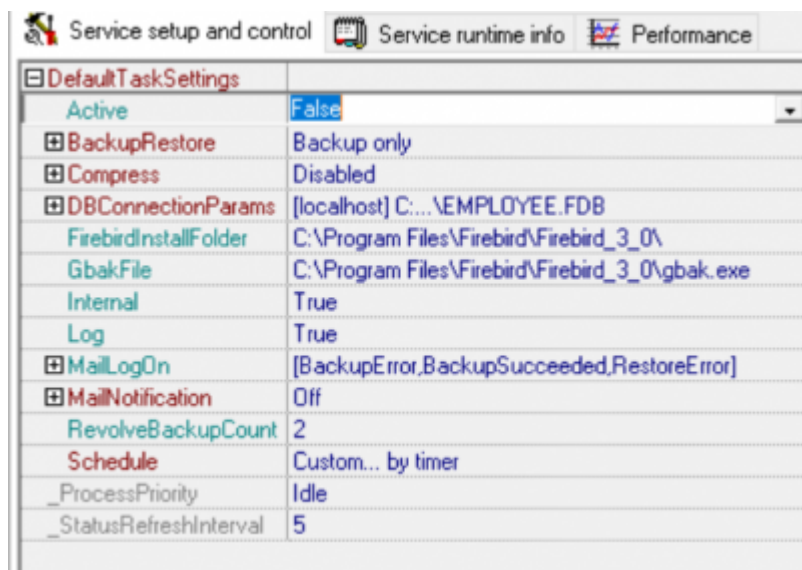


The following lists the various default settings and options available:

- Active
- Backup and Restore
- Compress
- Database connection configuration
- Firebird Install Folder
- Path to gbak.exe
- Logging
- Mail Notification
- Revolve Backup Count
- Schedule

After configuring the default task settings, all new tasks will have this configuration when created. It is of course possible to alter specific options for individual tasks.

back to top of page

# Default task settings

## Active

When True then the task just created will be active.

## Backup and Restore

This contains the basic backup and restore settings, processed by gbak.exe. Also there are few settings specific to the HK service, such as:

- **BackupFolder:** the folder where all backups will be stored
- **Restore/Enabled:** when *True*, then service will restore a database from a successful backup file. This can be used to validate the backup file.



If you need to perform any additional operations before/after the backup/restore (for example script execution, data validation, etc.) you may use the *Actions* options in the IBExpertBackupRestore service. The screenshot below shows the corresponding section with the *BeforeBackup* action expanded in SCC on the *Service setup* and *control* page.

Imagine that you've configured this task to backup a database my_server:c:\my_database.fdb and username and password are SYSDBA/masterkey. The *BeforeBackup* configuration example above means that before starting the database backup, the service will execute the command line:

```
C:\My programs\DataCheck.exe my_server:c:\my_database.fdb SYSDBA masterkey
```

If you need to interrupt the backup/restore process because some data validation or other operation has failed, you can use the InterruptOnFail option of the corresponding action. The execution of any action will be recognized as failed if the executed program sets the exit code not equal to 0 (zero).

The command line for each action may be configured using executable file parameters as well as with service macros. The macros will be replaced with corresponding values.

Here a description of the macros:

| Macro | Value |
|---|---|
| %database | Full connection string to source database. |
| %server | Database server name. |
| %database_file | Database file path. |
| %restored_database | Full connection string to restored database. |
| %backup_file | Path to backup file. |
| %role | SQL role from DBConnectionParams. |
| %user | Username from DBConnectionParams. |
| %password | Password from DBConnectionParams. |

To test the functionality of *Actions* you may use the special executable, DumpAction.exe, which only writes its command line to a log file (DumpAction.exe.log) and sets the exit code necessary. The exit code for this executable should be configured using a template such as:

```
DumpAction.exe -RESULT <integer_value>
```

For example, such a configuration of a *BeforeRestore* action will always stop the service performing the restore, because the exit code of such an action will be 2.

All actions with the corresponding results will be listed in the service report email message.

back to top of page

In the *Backup / Options* section you can configure the backup options, which are those available in GBAK, as required by simply setting the corresponding items to *True*. Please take into consideration that if you are intending to follow the backup with an immediate restore over the existing database, then the garbage collection makes no sense.



After that you will see the selected items in square brackets **[ ]** under *Backup / Options*,



and the corresponding gbak command-line parameters under *Backup*.



In the screenshots shown above you can see the backup configuration specified with the *No garbage collection* and *Ignore checksum* options.

When *Restore / Enabled* is set to True, the IBExpertBackupRestore restore will perform a restore from the backup just made. This feature can be useful if you want to validate the backup file or wish to use the freshly restored database for better performance.

*Restore / Info* can be used to execute up to 5 different queries, enabling you to obtain useful information about the status of the database, for example, the record count of a particular table, the last logged update timestamp or some special report. When *CollectInfo* is set to *True*, the restored database's main parameters, such as fle size, page size, pages count etc., can be viewed, or stored in the log file.

| BackupRestore | Backup and restore |
|---|---|
| Actions | BB AB BR AR |
| Backup | -IG -G -ZIP |
| General | -M |
| Restore | -REP -P 16384 |
| Enabled | True |
| GenerateBatFile | True |
| Info | 3 queries set. |
| CollectInfo | True |
| InfoQry1 | On (Top Sales Managers) |
| Enabled | True |
| Name | Top Sales Managers |
| SQL | (TStrings) |
| InfoQry2 | On (Employee count) |
| Enabled | True |
| Name | Employee count |
| SQL | (TStrings) |
| InfoQry3 | On (User relations list) |
| Enabled | True |
| Name | User relations list |
| SQL | (TStrings) |
| InfoQry4 | Off |
| InfoQry5 | Off |

In the *Restore / Settings* section you can set up the desired restore parameters, such as *restore folder*, *restore options*, *database page size*, etc. For example, if you want to restore a database from fresh backup into C:\My_Folder, create a database file, if no such file yet exists in the restore folder, or replace it if the file already exists. If you wish, you may also deactivate indices (DeactivateIndexes) to improve the performance of the restore. And perhaps you wish to re-specify the page size (PageSize) of the restored database to 16384. The screenshot below displays the corresponding *Restore/Settings* configuration:

| BackupRestore | Backup and restore |
|---|---|
| Actions | BB AB BR AR |
| Backup | -IG -G -ZIP |
| General | -M |
| Restore | -REP -CRE -INACTIVE -P 16384 |
| Enabled | True |
| GenerateBatFile | True |
| Info | 3 queries set. |
| Settings | -REP -CRE -INACTIVE -P 16384 |
| DeleteRestored | False |
| Options | [DeactivateIndexes,Replace,CreateNewDB] |
| FixFSSData | False |
| FixFSSMetada | False |
| DeactivateInd | True |
| NoShadow | False |
| NoValidityChec | False |
| OneRelationAt | False |
| Replace | True |
| CreateNewDB | True |
| UseAllSpace | False |
| ReadOnly | False |
| PageBuffers | 0 |
| PageSize | 16384 |
| RestoreFolder | C:\db\restore\ |
| UseAlternateServe | |

If you want make a restore just to validate a fresh backup file, you probably don't need to store the restored database file. So it is even possible to configure IBExpertBackupRestore to delete the restored database file following the restore. Just set the corresponding option to *True*.

| DeleteRestored | True |
| --- | --- |

[back to top of page](#)

## Restore to an alternative server

Backup and restore is very resource-consuming operation. To help your main database server breathe more easily, you can set the service to perform restores on an alternative server. This can be done using the *UseAlternateServer* option found in the *Restore* parameters.

| BackupRestore | Backup and restore |
| --- | --- |
| Actions | BB AB BR AR |
| Backup | -IG -G -ZIP |
| General | -M |
| Restore | localhost -REP -CRE -INACTIVE -P 16384 |
|   Enabled | True |
|   GenerateBatFile | True |
|   Info | 3 queries set. |
|   Settings | -REP -CRE -INACTIVE -P 16384 |
|   UseAlternateServe | localhost |
|     Enabled | True |
|     Server | localhost |
|       LoginParams | user:SYSDBA |
|         UserName | SYSDBA |
|         UserPasswo | ×××××××× |
|       Protocol | TCP |
|       ServerName | localhost |

When this option is enabled you can backup your database from one server and restore it to another.

## General Options

Firebird 3 introduced new backup options, so we have introduced a *General options* section, commensurate with the respective gbak switches:

```
gbak:Usage:
    gbak -b <db set> <backup set> [backup options] [general options]
    gbak -c <backup set> <db set> [restore options] [general options]
    <db set> = <database> | <db1 size1>...<dbN> (size in db pages)
    <backup set> = <backup> | <bk1 size1>...<bkN> (size in bytes = n[K|M|G])
    -recreate overwrite and -replace can be used instead of -c

gbak:legal switches are:
    -B(ACKUP_DATABASE)     backup database to file
    -C(REATE_DATABASE)     create database from backup file (restore)
    -R(ECREATE_DATABASE) [O(VERWRITE)] create (or replace if OVERWRITE used)
database from backup file (restore)
```

```
   -REP(LACE_DATABASE)     replace database from backup file (restore)

gbak:backup options are:
   -CO(NVERT)              backup external files as tables
   -E(XPAND)               no data compression
   -FA(CTOR)               blocking factor
   -G(ARBAGE_COLLECT)      inhibit garbage collection
   -IG(NORE)               ignore bad checksums
   -L(IMBO)                ignore transactions in limbo
   -NOD(BTRIGGERS)         do not run database triggers
   -NT                     Non-Transportable backup file format
   -OL(D_DESCRIPTIONS)     save old style metadata descriptions
   -T(RANSPORTABLE)        transportable backup -- data in XDR format

gbak:restore options are:
   -BU(FFERS)              override page buffers default
   -FIX_FSS_D(ATA)         fix malformed UNICODE_FSS data
   -FIX_FSS_M(ETADATA)     fix malformed UNICODE_FSS metadata
   -I(NACTIVE)             deactivate indexes during restore
   -K(ILL)                 restore without creating shadows
   -MO(DE) <access>        "read_only" or "read_write" access
   -N(O_VALIDITY)          do not restore database validity conditions
   -O(NE_AT_A_TIME)        restore one table at a time
   -P(AGE_SIZE)            override default page size
   -USE_(ALL_SPACE)        do not reserve space for record versions

gbak:general options are:
   -FE(TCH_PASSWORD)       fetch password from file
   -M(ETA_DATA)            backup or restore metadata only
   -PAS(SWORD)             Firebird password
   -RO(LE)                 Firebird SQL role
   -SE(RVICE)              use services manager
   -SKIP_D(ATA)            skip data for table
   -ST(ATISTICS) TDRW      show statistics:
       T                   time from start
       D                   delta time
       R                   page reads
       W                   page writes
   -TRU(STED)              use trusted authentication
   -USER                   Firebird user name
   -V(ERIFY)               report each action taken
   -VERBI(NT) <n>          verbose information with explicit interval
   -Y   <path>             redirect/suppress status message output
   -Z                      print version number

gbak:switches can be abbreviated to the unparenthesized characters
```

In *General options* you can set parameters such as *fetch password from file, use service, use trusted authentication, backup metadata only* etc. as shown in below:

| BackupRestore | Backup and restore |
|---|---|
| Actions | BB AB BR AR |
| Backup | -IG -G -ZIP |
| General | -M |
| FetchPassword | |
| Options | [MetadataOnly] |
| MetadataOnly | True |
| Trusted | False |
| OtherOptions | |
| SkipData | |
| UseService | |
| Verify | False |
| Restore | localhost -REP -CRE -INACTIVE -P 16384 |

[back to top of page](#)

**Compress**

If you want to compress a successfully created backup file, you should use this configuration section. You can also configure the service here to delete the backup file, following the successful compression (DeleteBackupAfterCompress option).

To make the backup compression work you should set Enabled to True, and then configure the appropriate compress settings. You can use the built-in ZIP compressor or configure the service to run an external compressor exe file. Here is a screenshot of the compress settings configured to use the built-in ZIP compressor:

| Compress | Built in ZIP |
|---|---|
| Enabled | True |
| Settings | |
| CompressByBuiltInZ | True |
| CompressCommand | |
| CompressedFileExt | zip |
| DeleteBackupAfter | True |
| ExecutableFile | |

Here is a screenshot of a configuration using an external compressor (for example WinRAR):

| Compress | Custom. Into rar-file |
|---|---|
| Enabled | True |
| Settings | |
| CompressByBuiltInZIP | False |
| CompressCommandLine | a %compressed %backup |
| CompressedFileExt | rar |
| DeleteBackupAfterCompress | True |
| ExecutableFile | C:\Program Files\WinRAR\Rar.exe |

The *CompressCommandLine* option can contain three macros, which will be replaced with the corresponding values when calling the compressor:

| %backup | Backup file name with extension. |
|---|---|

| %compressed | Compressed file name = backup file name + extension. |
| %back_filename | Backup filename without extension. |

The extension is configured in *CompressedFileExt*.

[back to top of page](#)

**Database connection configuration**

The essential key to any database manipulation (except moving it into the recycler!) is establishing the database connection. All necessary properties can be configured in the *DBConnectionParams* section:

| ⊟DBConnectionParams | [localhost] C:\Firebird 30 Win32\fb3w32\Restore\restore.fdb |
|---|---|
| CharactersSet | |
| DatabaseName | C:\Firebird 30 Win32\fb3w32\Restore\restore.fdb |
| ⊟LoginParams | user:SYSDBA |
| UserName | SYSDBA |
| UserPassword | ******** |
| Protocol | TCP |
| ServerName | localhost |
| SqlRole | |

You can also add a role; the SYSDBA user and masterkey password are the default.

This is fairly self-explanatory, although should you require detailed information regarding Firebird/InterBase® database connection parameters, please refer to the online [IBExpert documentation](#).

[back to top of page](#)

**Firebird Install Folder**

IBExpertBackupRestore implements a so-called internal function for performing the back/restore. So you need to let the service know where Firebird is installed:

| FirebirdInstallFolder | C:\Program Files\Firebird\Firebird_3_0\ |

**Path to gbak.exe**

IBExpertBackupRestore collaborates with gbak.exe to enhance the backup/restore tasks. So you need to let the service know where this file can be found:

| GbakFile | C:\Program Files\Firebird\Firebird_3_0\gbak.exe |

We use gbak because it has all the functionalities from the newest Firebird version.

## Logging

It's likely you'd like to have log files of your backup/restore operations. Those files may help you to understand what's wrong with your database, should an error occur during the backup/restore process. To enable such log files, just set the corresponding option to *True*, as shown below:

| Log | True |
| --- | --- |

back to top of page

## Mail notification

You may use the mail notification feature if you want to receive reports about IBExpertBackupRestore's activity. The service sends an email message with log files attached when the backup/restore task is completed.
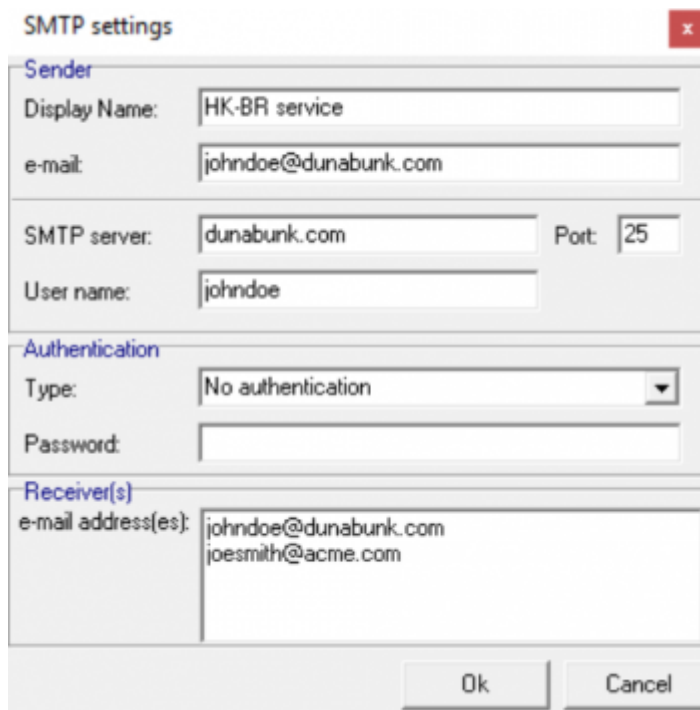
The *MailLogOn* option is used to define the situations, when log files should be mailed. For example, if you'd like to receive log files when a backup or restore has failed, you should specify the options as follows:

| MailLogOn | [BackupError,RestoreError] |
| --- | --- |
| BackupError | True |
| BackupSucceeded | False |
| RestoreError | True |
| RestoreSucceeded | False |

To use the mail notification feature, the *Enabled* parameter in the *MailNotification* section should be set to *True*.

| MailNotification | On |
| --- | --- |
| Enabled | True |
| SmtpSettings | 2 receivers |

IBExpertBackupRestore uses a built-in SMTP client to send emails, so you need to set up the SMTP parameters in the task configuration to enable this to work properly. Simply double-click on the *SmtpSettings* option, to open the configuration dialog window.

In this dialog you should set up the *Sender*, *SMTP server configuration* and one or more recipients.

[back to top of page](#)


**Revolve backup count**


IBExpertBackupRestore works as a rotator when creating a new backup. If a new backup is successfully created, the oldest one will be deleted. Such mechanics let you configure the service to store just *n* last backups. The *n* value can be configured using the *RevolveBackupCount* option:



[back to top of page](#)
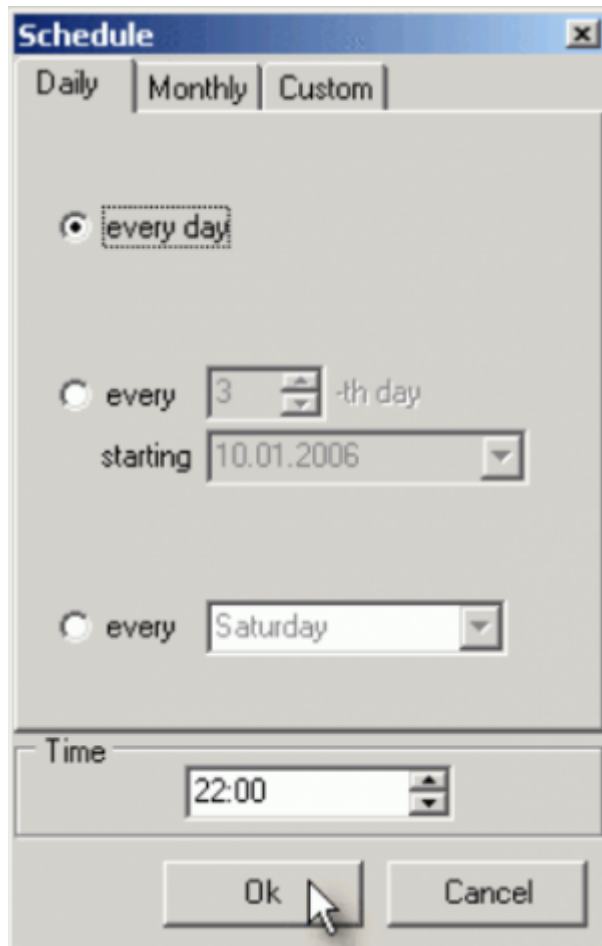

**Schedule**



Double-click on the *Schedule* option to open the schedule configuration dialog window:
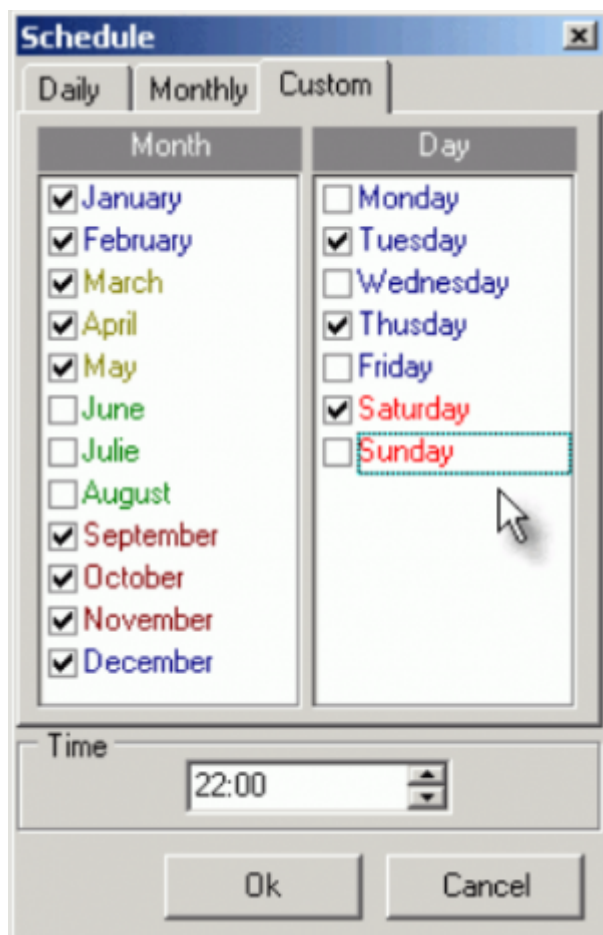
**Daily schedule:**

- every day at the specified time.
- every *n*th day, starting from date.
- every given day of week.

**Monthly schedule:**
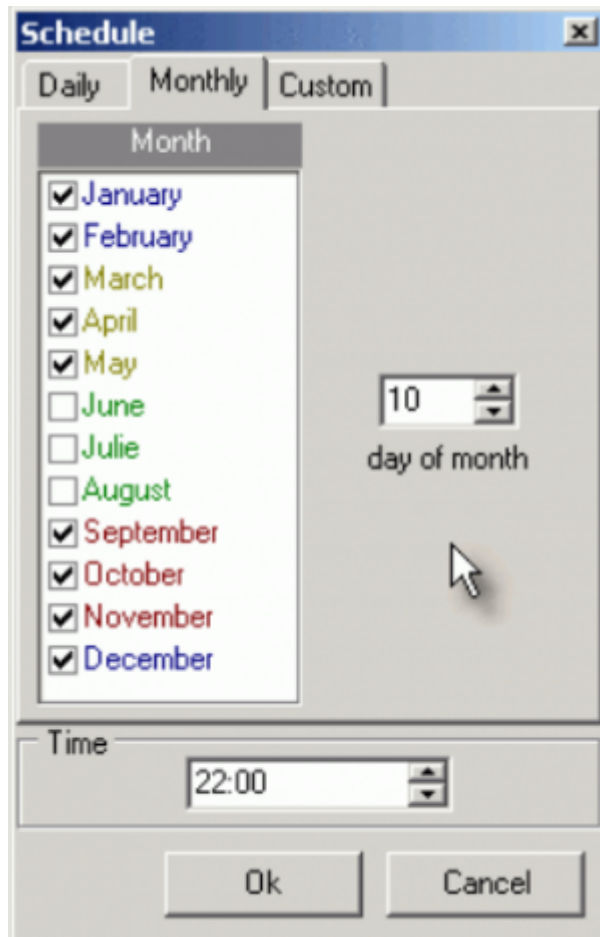
Every *n*th day of the selected months at the given time.

**Custom schedule:**

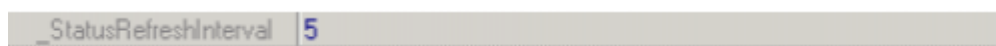Selected days of every week of selected months at given time.

### _ProcessPriority

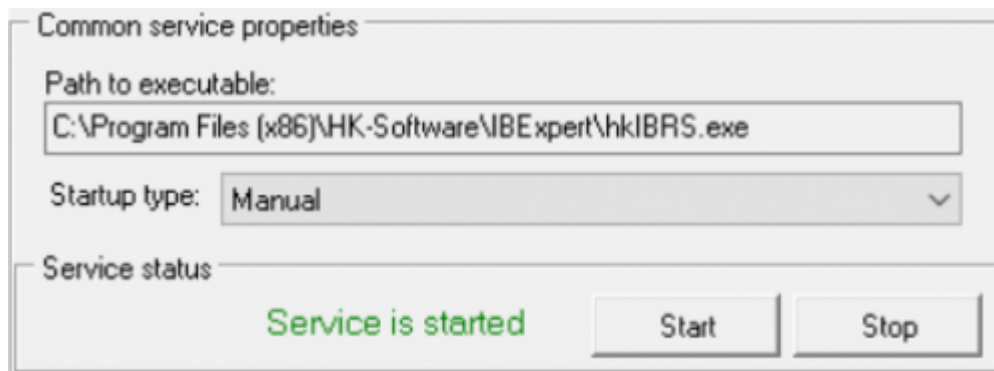This parameter can be set to *Idle*, *Normal* or *High* (the default is *Idle*).

### _StatusRefreshInterval

Here the refresh interval in seconds can be specified (default value is 5).

# Common service properties

The path to the executable file, hkIBRS.exe is displayed. You can specify the *Startup type* selecting an option from the drop-down list (options: *Manual*, *Automatic* or *Disabled*).

The *Service Status* can be viewed at the bottom of the window, and the *Start* and *Stop* buttons used to manually start or stop the service.

Once you are sure you've configured your default settings as you need them, don't forget to save your configuration by clicking the disk icon in the toolbar, before moving on to configuring your individual databases for their backup.