Database Shadow

Shadow files are an exact live copy of the original active database, allowing you to maintain live duplicates of your production database, which can be brought into production in the event of a hardware failure. These shadows are administrated in real time by the Firebird/InterBase® server. They are used for security reasons: should the original database be damaged or incapacitated by hardware problems, the shadow can immediately take over as the primary database. It is therefore important that shadow files do not run on the same server or at least on the same drive as the primary database files.

Firebird/InterBase® allows up to 65,536 (216) database files, including shadow files. However the operating system used may have a lower limit on the number of simultaneous open files that the Firebird/InterBase® server can have. In some cases, the operating system provides a means to raise this limit (refer to your OS documentation for the default open files limit, and the means to raise it).

Shadow files, as with the main database and secondary files, may not reside on networked or remote file systems (i.e. mapped drives on Windows and NFS files on UNIX).

The number of existing shadow files in a database may be ascertained using the IBExpert Services menu item, Database Statistics, or using GSTAT (the shadow count is included in the database header page information).

Shadowing offers a number of advantages:

- It provides valuable protection of the database, in addition to the regular backups which should be maintained, and in addition to Firebird/InterBase®'s multigenerational architecture.
- If the original database is damaged, the shadow can be activated immediately, with little lost time.
- Shadowing runs automatically with little or no maintenance.
- You have full control over the shadow's configuration, including its use of hard disk space and distribution across other available devices.
- Creating a shadow does not require exclusive access to the database.
- Shadow files use the same amount of disk space as the database, as opposed to log files, which can grow well beyond the size of the database.
- Shadowing does not use a separate process. The database process handles writing to the shadow.

But there are also some limitations:

- Shadowing only helps to recover from certain types of problems. If a user error or Firebird/InterBase® problem causes the database to be damaged beyond recovery, then the shadow is identically damaged. But if the database is accidentally deleted by the user, or a hardware problem on the primary server occurs, the shadow remains intact and can be used immediately.
- Shadowing is not replication. It is one-way writing, duplicating every write operation on the master database. Client applications cannot access the shadow file directly.
- The shadow cannot be used to rollback the database to a specific point in time. When the shadow is used to recover the database, everything up to the point where the original problem occurred is retrieved.
- Shadowing adds a small performance penalty to database operations. Every action on the

database which modifies metadata or the data itself is mirrored in the shadow.
- Shadowing does not replace a careful security system within the operating system, but is one aspect or enhancement of the whole.
- Shadowing also works only for operations that go through the Firebird/InterBase® database services manager (GDS), which processes all SQL and database requests.
- Shadowing can occur only to a local disk. Shadowing to a NFS file system or mapped drive is not possible.
- Shadowing to tape or other media is also not possible.

Further information regarding shadow usage can be referred to the Firebird administration using IBExpert documentation chapter, Working with shadows.

back to top of page

Tasks for shadowing

The main tasks in setting up and maintaining shadows are as follows:

Creating a shadow

(Source: InterBase® 7.1 Operations Guide)

Shadowing begins with the creation of a shadow, using the CREATE SHADOW statement. This statement has the following syntax:

```
CREATE SHADOW shadow_number
[AUTO | MANUAL] [CONDITIONAL] shadow_filename
```

The shadow number identifies a shadow set that collects the primary shadow file and any secondary files together. The most important function of the shadow number is to identify the shadow if you decide to drop it (please refer to Deleting a shadow).

This can be performed without affecting users at all, as it does not require exclusive access. Before creating the shadow, the following should be considered:

- **Shadow location:** a shadow should be created on a different disk from the main database, as shadowing is intended as a recovery mechanism in case of disk failure. Therefore storing the main database and the shadow on the same disk defeats the whole purpose of shadowing!
- **Distributing the shadow:** a shadow can be created as a single-file (shadow file) or as multiple files (shadow set). To improve space allocation and disk I/O, each file in a shadow set may be placed on a different disk.
- **User access:** if a shadow becomes unavailable, user access to the database can be denied until shadowing is resumed, or access can be allowed (i.e. work can continue as normal) although any changes made during this period will obviously not be shadowed. Please refer to auto mode and manual mode for further information.
- **Automatic shadow creation:** To ensure that a new shadow is automatically created, create a

conditional shadow (details below).

Please note: If the IBExpert Services menu item Restore Database option, *Don't Recreate Shadow Files* is checked, shadow files are not recreated while restoring. This deletes the shadow definition; and to restore it, it is necessary to recreate the shadow using the CREATE SHADOW statement. This option is sometimes required if the destination database does not support shadows, if you are migrating from an earlier version of InterBase® where shadows are not supported, or if the machine where the shadow resides is not available.

The following sections deal with the creation of shadows with various options:

- Creating single-file or multifile shadows
- Auto mode and manual mode
- Conditional shadows

These options are not mutually exclusive, e.g. it is possible to create a single-file conditional shadow with the option manual mode.

Creating single-file or multifile shadows

(Source: InterBase® 7.1 Operations Guide)

To create a single-file shadow for the sample database employee.gdb, enter the following in the IBExpert SQL Editor:

```
CREATE SHADOW 1 '/usr/interbase/examples/employee.shd';
```

The name of the shadow file is employee.shd, and it is identified by the number 1. It is possible to verify that the shadow has been created by using the isql command:

```
SHOW DATABASE;
   Database: employee.gdb
   Shadow 1: '/usr/interbase/examples/employee.shd' auto
   PAGE_SIZE 4096
   Number of DB pages allocated = 392
   Sweep interval = 20000
```

The page size of the shadow is the same as that of the database. A large database may be shadowed to a multifile shadow if wished, spreading the shadow files over several disks. Each file in the shadow set needs to be specified by name and size. This can be specified in two ways, the same as with multifile databases:

- Specify the page on which each secondary file starts
- Specify the length in database pages of each file.

You can specify both but this is redundant. If the information specified is inconsistent, Firebird/InterBase® uses the length value in preference to the starting page value. In general, it is best to use either length values or starting page number to ensure consistency or legibility.

If the files are specified using the LENGTH keyword, do not specify the length of the final file, as Firebird/InterBase® sizes the final file dynamically, as needed. Please refer to Secondary Files

Manager for further information.

The following example creates a shadow set consisting of three files. The primary file, EMPLOYEE.SHD, is 10,000 database pages in length; the second file is 20,000 pages long, and the final file is left open, to expand as needed.

```
CREATE SHADOW 1 'employee.shd' LENGTH 10000
   FILE 'emp2.shd' LENGTH 20000
   FILE 'emp3.shd';
```

The second alternative is to specify the starting page of the files:

```
CREATE SHADOW 1 'employee.shd'
   FILE 'emp1.shd' STARTING AT 10000
   FILE 'emp2.shd' STARTING AT 30000;
```

Using the SHOW DATABASE command, the file names, page lengths or starting pages can be verified:

```
SHOW DATABASE;
   Database: employee.gdb
   Shadow 1: '/usr/interbase/examples/employee.shd' auto length 10000
   file /usr/interbase/examples/emp1.shd length 2000 starting 10000
   file /usr/interbase/examples/emp2.shd length 2000 starting 30000
   PAGE_SIZE 4096
   Number of DB pages allocated = 392
   Sweep interval = 20000
```

The page length for secondary files in the main database does not need to correspond to the page length for the secondary shadow files. As the database grows and its first shadow file becomes full, updates to the database automatically overflow into the next shadow file.

Auto mode and manual mode

(Source: InterBase® 7.1 Operations Guide)

A shadow database may become unavailable for the same reasons a database becomes unavailable (e.g. disk failure, network failure, or accidental deletion). If a shadow has been created in auto mode and suddenly becomes unavailable, database operations continue automatically without shadowing. If the shadow was created in manual mode, further access to the database is denied until the database administrator gives explicit instructions, as to how work is to be continued.

The benefits of auto mode and manual mode may be compared below:

Mode	Advantage	Disadvantage
Auto	**Database operation is uninterrupted.**	Creates a temporary period when the database is not shadowed. The database administrator might be unaware that the database is operating without a shadow.

Mode	Advantage	Disadvantage
Manual	**Prevents the database from running unintentionally without a shadow.**	Database operation is halted until the problem is fixed. Needs intervention of the database administrator.

Auto mode

The AUTO keyword can be used to create a shadow in auto mode:

```
CREATE SHADOW 1 AUTO 'employee.shd';
```

Auto mode is the default, so this does not necessarily need to be specified explicitly.

In auto mode, database operation is uninterrupted even though there is no shadow. To resume shadowing, it might be necessary to create a new shadow. If the original shadow was created as a conditional shadow, a new shadow is automatically created. Please refer to Conditional shadows for further information.

Manual mode

The MANUAL keyword can be used to create a shadow in manual mode:

```
CREATE SHADOW 1 MANUAL 'employee.shd';
```

Manual mode is useful when continuous shadowing is more important than continuous operation of the database. When a manual-mode shadow becomes unavailable, further operations on the database are prevented.

To allow work on the database to be resumed, the database owner or SYSDBA must enter the following command:

```
gfix -kill database
```

This command deletes metadata references to the unavailable shadow corresponding to the database. After deleting the references, a new shadow can be created if shadowing needs to be resumed.

Shadow information is kept in the metadata of the primary database file. If this file becomes unavailable for some reason, then the pointers to the shadow are also broken. In this situation, the database administrator can use the -active option in the GFIX utility to convert the original shadow into a new primary database.

Conditional shadows

(Source: InterBase® 7.1 Operations Guide)

A shadow may be defined so that if it replaces a database, the server creates a new shadow file, and thus allows shadowing to continue uninterrupted. This is termed a conditional shadow, and is specified using the CONDITIONAL keyword:

```
CREATE SHADOW 3 CONDITIONAL 'atlas.shd';
```

Creating a conditional file automatically creates a new shadow in either of two situations:

- The database or one of its shadow files becomes unavailable.
- The shadow takes over for the database due to hardware failure.

back to top of page

Activating a shadow

(Source: InterBase® 7.1 Operations Guide)

Should the main database become unavailable for whatever reason, the shadow can be activated, i.e. it takes over the main database and all users now access the shadow as the main database. This activation may be defined to occur automatically or through the intervention of the database administrator.

Shadow information is kept in the metadata of the primary database file. If this file becomes unavailable for some reason, then the pointers to the shadow are also broken. To activate the shadow it is necessary to log in as SYSDBA or the database owner, and use GFIX with the -activate option, to convert the original shadow into a new primary database.

Important! The first step is to make sure the shadow is not active, i.e. if the main database has active transactions the shadow is active. Also check that the main database is unavailable. If a shadow is activated while the main database is still available, the shadow can be corrupted by existing attachments to the main database.

To activate a shadow, specify the path name of its primary file:

```
gfix -a[ctivate] shadow_name
```

shadow_name is the explicit path and name of the shadow's primary file.

Examples

For a Windows NT server:

```
gfix -a F:\SHADOW\ORDENT\ORDERS.SHD
```

For any UNIX server:

```
gfix -a /usr/shadow/ordent/orders.shd
```

After a shadow is activated its name should be changed to the name of the original database. Then a new shadow can be created if shadowing needs to continue providing another disk drive is available.

back to top of page

Deleting a shadow

(Source: InterBase® 7.1 Operations Guide)

If a shadow is no longer needed, it can be stopped by simply deleting it. To stop shadowing, use the shadow number as an argument with the DROP SHADOW statement. For example:

```
DROP SHADOW 1
```

If you need to look up the shadow number, use the isql command SHOW DATABASE.

Important! DROP SHADOW deletes all shadow references from a database's metadata as well as the physical files on disk. Once the files have been removed from the disk, there is no way to recover them. However, as a shadow is merely a copy of an existing database, a new shadow will be identical to the dropped shadow.

A shadow can be dropped by its creator, the SYSDBA user, or any user with operating system root privileges.

back to top of page

Adding files to a shadow/modifying a shadow

(Source: InterBase® 7.1 Operations Guide)

Shadow databases may consist of multiple files. As the shadow grows in size, files may need to be added to cope with the increase in space requirements.

To modify a shadow database or add a shadow file, first use the DROP SHADOW statement to delete the existing shadow, then use the CREATE SHADOW statement to create a multifile shadow.

Example

```
DROP SHADOW 2
CREATE SHADOW 3 AUTO CONDITIONAL
'F:\SHADOW\ORDENT\ORDERS.SHD' LENGTH 10000
FILE 'F:\SHADOW\OIRDENT\ORDERS2.SHD'
```

The page length allocated for secondary shadow files need not correspond to the page length of the database's secondary files. As the database grows and its first shadow file becomes full, updates to the database automatically overflow into the next shadow file.

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

Permanent link:
https://www.ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:database-shadow

Last update: **2023/08/14 15:05**