

# 25

## Backup and Restoring of the Database

### Introduction

4D includes a full database backup and restore module.

This module allows backing up a database currently in use without having to exit it. Each backup can include the structure file, the data file and any additional files or folders. These parameters are first set in the application Preferences.

Backups can be started manually or automatically at regular intervals without any user intervention. Specific language commands, as well as specific database methods, allow integrating backup functions into a customized interface.

Databases can be restored automatically when a damaged database is opened.

Also, the integrated backup module can take advantage of the log file. This file keeps a record of all operations performed on the data and also ensures total security between two backups. In case of problems with a database in use, any operations missing in the data file are automatically reintegrated the next time the database is opened. You can view the log file contents at any time.

The integrated backup module allows you to:

- Start a complete backup of database files at any time (structure file, data file, log file, attached files, etc.),
- Set up automatic backups at regular intervals — on a hourly, daily, weekly or monthly basis,

- Set advanced parameters for backups (number of sets, file compression, options for startup after a restore, etc.),
- Automatically restore a database and its attached files in case of incident,
- Automatically integrate missing operations stored in the log file into a restored database,
- Roll back operations performed on database data.

## Managing Backups

Backing up the database consists of generating a copy of the database and all its necessary files at a given moment. This copy is placed in one or more backup file(s).

Any backup file can later be opened using 4D in case of any incidents that damage the current database; the database will then be restored to its previous state (at the time it was copied).

### Performing a Backup

Each backup is performed while taking into account the parameters set in the application Preferences.

The Preferences can be used to define every aspect of the backup:

- Files to include in the backup (data / log, structure, user structure and attached files),
- Location of backup files (main backup file and log backup file).
- Management of log file,
- Backup scheduling,
- Backup options: number and rotation of backup sets; handling of active transactions or index operations; handling of failures; segmentation, compression and integrity of backup files,
- Setting automatic restore options.

These parameters are set at default values corresponding to standard use; changing these values is optional. For more information on these parameters, please refer to the [“Configuration of Backup Files” paragraph on page 1214](#) and the [“Backup Settings” paragraph on page 1219](#).

## Starting a Backup

In 4D, a backup can be started in three ways:

- Manually, using the **Backup...** command in the 4D or the **Backup** button of the Maintenance and Security Center (MSC).
- Automatically, using the scheduler that can be set in the application Preferences.
- Programmatically, using the BACKUP command.

The choice will depend on your use of the database and your backup strategy.

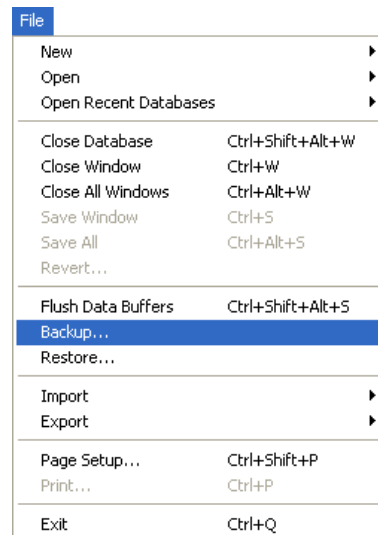
---

*4D Server* A backup can be started manually from a remote machine using a method that calls the BACKUP command. The command will be executed, in all cases, on the server.

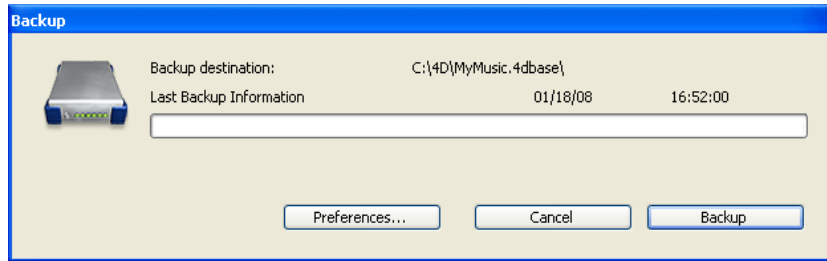
---

- To perform a manual backup:

### 1 Select the **Backup...** command in the 4D **File** menu:

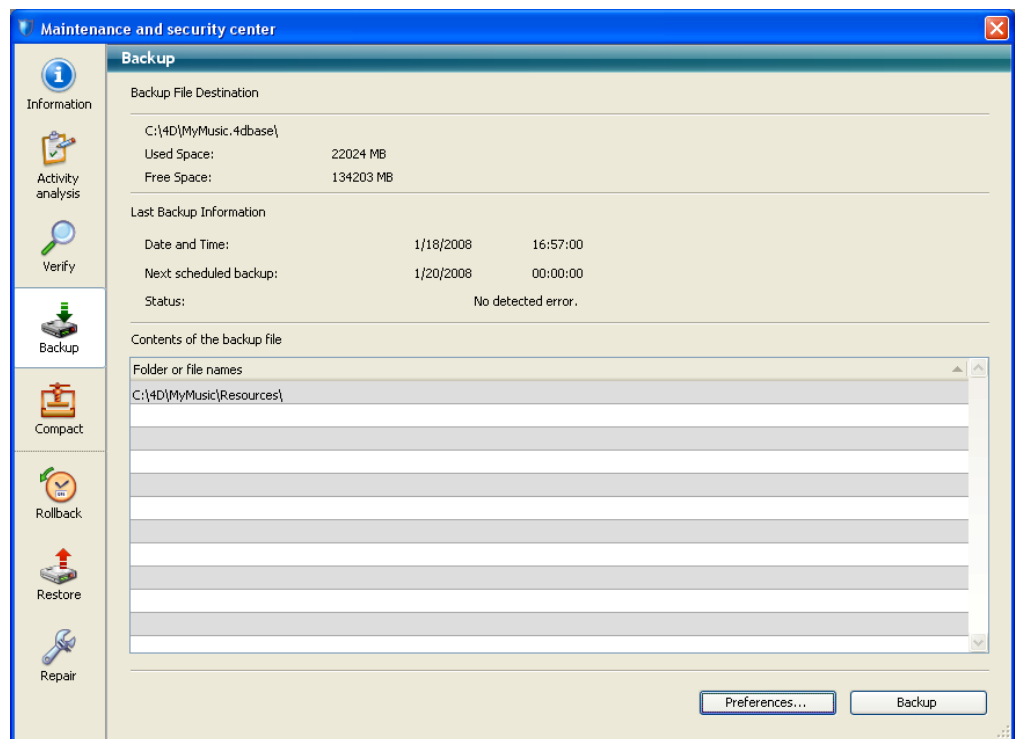


The backup window appears:



OR

Select **Maintenance Security Center** in the **Help** menu of 4D and display the “Backup” page.



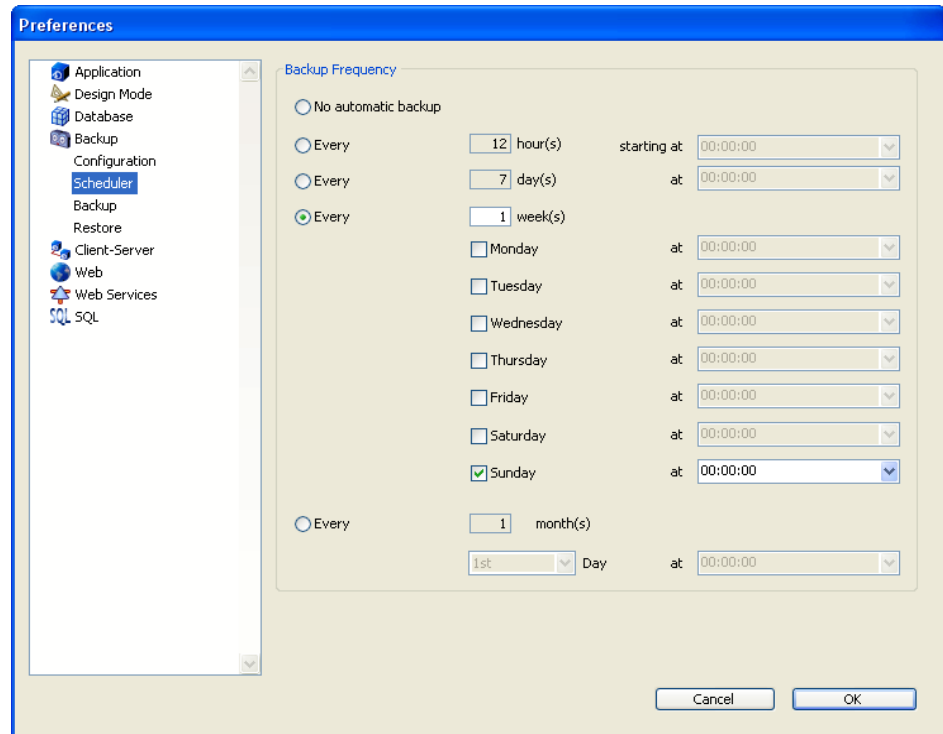
For more information about the MSC, please refer to the [chapter “Maintenance and Security Center”](#) on page 1247.

The **Preferences** button causes the general backup preferences to be displayed.

**2** Click **Backup** to start the backup using current parameters.

- To perform a scheduled automatic backup:

- 1 In the **Scheduler** page of the “Backup” Preferences, set the backup frequency:



Backups are automatically performed at the times defined on this page without any type of user intervention.

*Note* For more information on using this dialog box, please refer to the “[Scheduled Backup Settings](#)” paragraph on page 1223.

- To perform a scheduled backup using 4D language:

- 1 **Execute the BACKUP command in a method.**

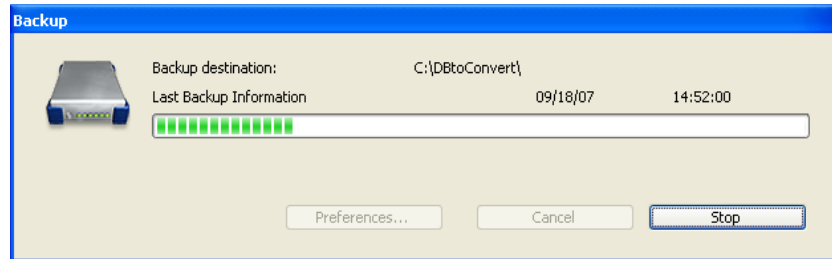
The backup starts using the current parameters.

You can use the On Backup Startup and On Backup Shutdown database methods for handling the backup process.

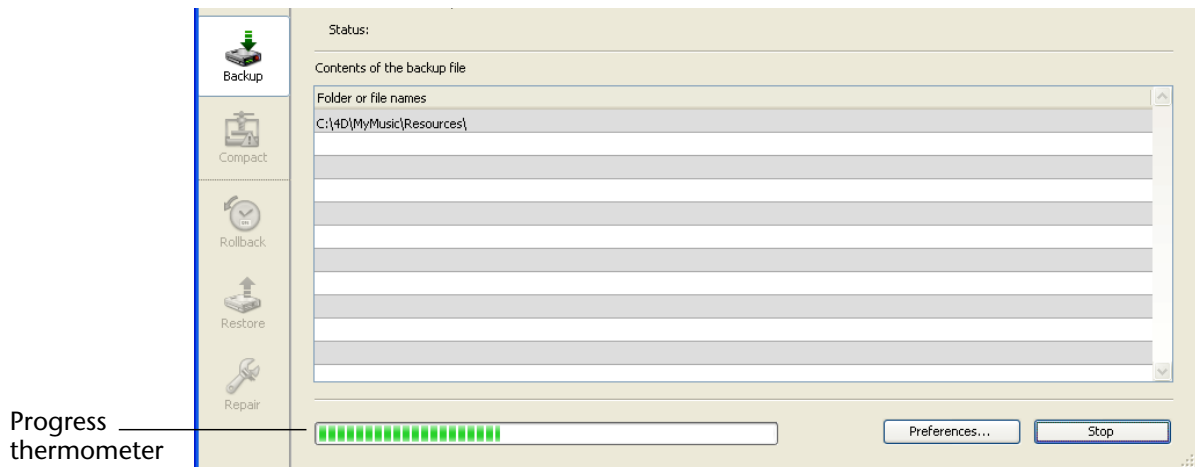
For more information on this, please refer to the *4D Language Reference* manual.

### Executing a Backup

Once the backup is started, 4D displays a dialog box with a thermometer indicating the progress of the backup:



This thermometer is also displayed on the “Backup” page of the MSC if you have used this dialog box:



The **Stop** button lets the user interrupt the backup at any time (please refer to the [“If Backup Fails” paragraph on page 1221](#)).

The result of the last backup (successful or failed) is stored in the “Last Backup Information” area of the **Backup/Configuration** page of the Preferences or in the main window of 4D Server (please refer to the [“Last Backup Information” paragraph on page 1218](#)). It is also recorded in the database Backup journal (please refer to the [“Backup Journal” paragraph on page 1244](#)).

### Accessing the Database During Backup

During a backup, access to the database is restricted by 4D according to the context.

4D locks any processes related to the types of files included in the backup: if only the structure file is being backed up, access to the structure is not possible but access to the data will be allowed.

Conversely, if only the data file is being backed up, access to the structure is still allowed. In this case, the database access possibilities are as follows:

- With the 4D single-user version, the database is locked for both read and write; all processes are frozen. No actions can be performed.
- With 4D Server, the database is only write locked; client machines can view data. If a client machine sends an add, remove or change request to the server, a window appears asking the user to wait until the end of the backup.

Once the database is saved, the window disappears and the action is performed. To cancel the request in process and not wait for the end of the backup, simply click the **Cancel operation** button.

However, if the action waiting to be executed comes from a method launched prior to the backup, you should not cancel it because only operations remaining to be performed are cancelled. Also, a partially executed method can cause logical inconsistencies in the database.

---

*Note* When the action waiting to be executed comes from a method and the user clicks the **Cancel operation** button, 4D Server returns error -9976 (This command cannot be executed because the database backup is in progress).

---

### Encountering Problems During a Backup

It may happen that a backup is not executed properly. There may be several causes of a failed backup: user interruption, attached file not found, destination disk problems, incomplete transaction, etc. 4D processes the incident according to the cause.

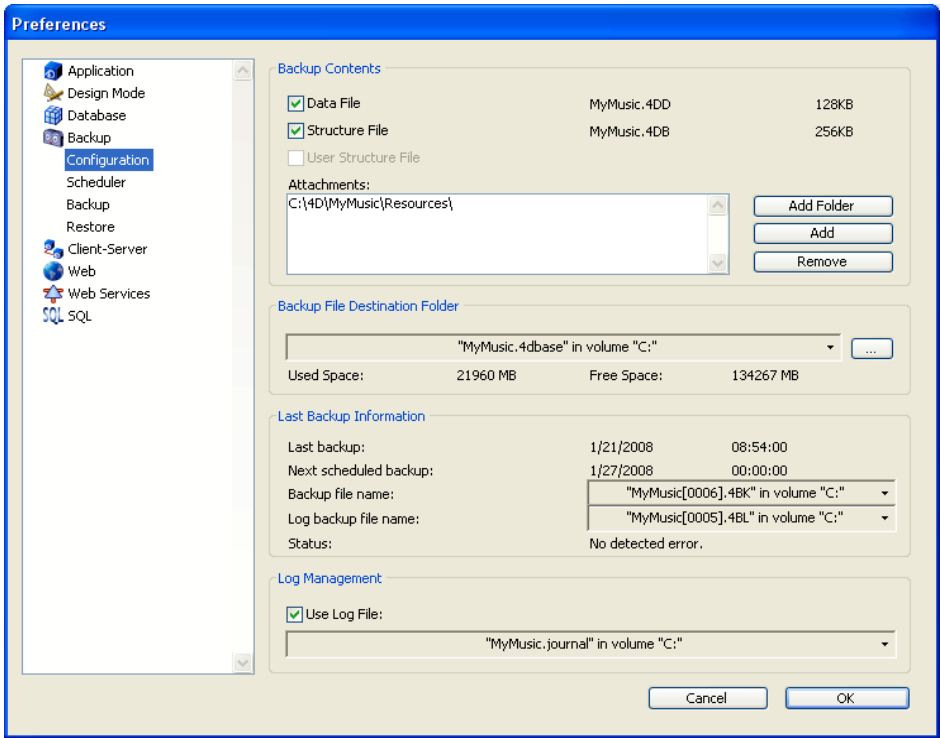
In all cases, the status of the last backup (successful or failed) is displayed on the **Configuration** page in the **Backup** preferences, in the 4D Server window and in the Backup journal (please refer to the [“Last Backup Information”](#) paragraph on page 1218).

- **User interruption:** The **Stop** button in the progress dialog box allows users to interrupt the backup at any time. In this case, the copying of elements is stopped and an error is generated. You can intercept this error in the On Backup Shutdown database method.
- **Attached file not found:** When an attached file cannot be found, 4D performs a partial backup (backup of database files and accessible attached files) and returns an error.

- **Backup impossible** (disk is full or write-protected, missing disk, disk failure, incomplete transaction, database not launched at time of scheduled automatic backup, etc.):
  - If this is a first-time error, 4D will then make a second attempt to perform the backup. The wait between the two attempts is defined on the **Backup** page of the Preferences (please refer to the “If Backup Fails” paragraph on page 1221).
  - If the second attempt fails, a system alert dialog box is displayed and an error is generated. You can intercept this error in the On Backup Shutdown database method.

Configuration of Backup Files

The **Configuration** page of the application Preferences lets you set the backup files and their location, as well as that of the log file. It also provides information on the last backup:



These parameters are specific to each database opened by the 4D application.

*4D Server* These parameters can only be set from the 4D Server machine.



## Backup Contents

This area allows you to define which files and/or folders to copy during the next backup.

The upper portion of the area lists the 4D database files and indicates their current size. You must set each file to include in the backup by checking the corresponding option. A dimmed option means that the corresponding file is not available in the database. You can select the 4D files that you want, depending on how often they are updated, their strategic interest, their size, etc. No file is required.

The lower portion lists the file access paths of any attached files in the backup.

- **Data File:** Database data file.

When this option is checked, the **index file** (.4DIndx) as well as the current **log file** of the database, if any, is backed up at the same time as the data. The backup causes the closing and backup of the current log file, then the creation of a new log file. This prevents the size of the log file from becoming excessively large.

For more information on the log file, please refer to the [“Managing the Log File” paragraph on page 1226](#).

- **Structure File:** Database structure file.

In cases where databases are compiled and merged with 4D Unlimited Desktop, this option allows you to backup the .exe file (Windows) and the package (Mac OS).

- **User Structure File** (optional): Database User structure file that contains customized user forms (if any). For more information about user forms, please refer to the [chapter “User Forms” on page 705](#).

- **Attachments:** This area allows you to specify a set of files and/or folders to be backed up at the same time as the database. These files can be of any type (documents or plug-in templates, labels, reports, pictures, etc.).

You can set either individual files or folders whose contents will be fully backed up. Each attached element is listed with its full access path in the “Attachments” area.

- **Add Folder:** When you click this button, 4D displays a dialog box that allows selecting a folder to add to the backup. In case of a restore, the folder will be recuperated with its internal structure. You can select any folder or volume connected to the machine, with the exception of the folder containing the database files.

- **Add:** When you click this button, 4D displays a dialog box that allows selecting a file to add to the backup. You cannot select a database file as an attached file.
- **Remove:** This button allows you to remove the selected file from the list of attached files.

*Note* It is possible that one or more attached files are not accessible when the backup is executed (modified name or access path, disk disconnected, etc.). In this case, the backup is executed without the missing file(s) and an error is generated. You can intercept this error in the On Backup Shutdown Database Method. The error is also indicated in the Backup journal.

### Backup File Destination Folder

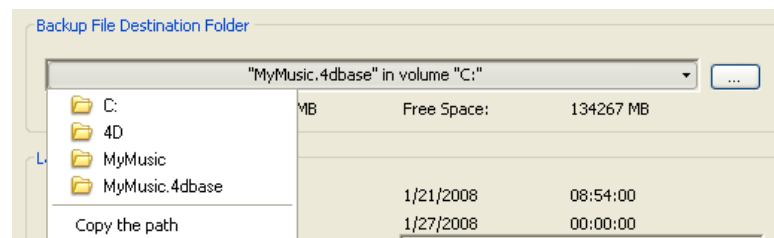
This area allows viewing and modifying the location where backup files and log files (if any) are stored.

4D generates two types of backup files: backup files and backup log files. Backup files are special for two reasons:

- They can contain several files (structure, data, attached files, etc.).
- They are secured using internal verification mechanisms (which can be set on the “Backup” page of the Preferences).

By default, 4D stores these files next to the database data file. It is strongly advised to set a location on another disk volume to reduce the risk of data loss in case of disk failure on the drive containing the database.

You can click in the area indicating the location of the backup files in order to display their pathname as a pop-up menu containing the series of folders on the disk:



If you select an item in this menu, it will be displayed in a new system window. The **Copy the path** command copies the complete pathname as text in the clipboard.

To modify the location where these files are stored, click the [...] button. A selection dialog box appears, which allows selecting a folder or disk where the backups will be placed. The “Used Space” and “Free Space” areas are updated automatically and indicate the remaining disk space on the disk of the selected folder.

You should make sure that the free space is sufficient for all of your backups. If a backup fails due to a lack of disk space, an error is generated. You can intercept this error in the On Backup Shutdown Database Method. The error is also indicated in the information area and in the Backup journal.

## Backup File Names

4D names backup files using a specific naming system on which the automatic restore functions are based. This naming system cannot be changed.

Standard backups are named ***Databasename[xxxx].4BK***, where *databasename* is the name of the database data file and *xxxx* is the number of the backup. For example, the 26<sup>th</sup> backup of the Invoices database is named *Invoices[0026].4BK*.

If the backup is segmented, 4D adds the segment number as -*xxxx*. For example, the 3<sup>rd</sup> segment of the 26<sup>th</sup> backup of the Invoices database is named *Invoices[0026-0003].4BK*.

For more information on segments, please refer to the [“Archive” paragraph on page 1222](#).

- Backups of log files are named ***Logname[xxxx].4BL***, where *logname* is the name of the log file of the database and *xxxx* is the number of the backup (starting at 0). For example, the 13<sup>th</sup> backup of the Log log file is named *Log[0012].4BL*.

If the backup of the log file is segmented, 4D adds the segment number as -*xxxx*. For example, the 2<sup>nd</sup> segment of the 13<sup>th</sup> backup of the log file Log is named *Log[0012-0002].4BL*.

For more information on segments, please refer to the [“Archive” paragraph on page 1222](#).

Please note that log file backups start at 0 while database file backups start at 1. For the first database file backup (backup[0001].4BK for example), the log file backup is named log[0000]: it represents the changes made in the data file starting at its “empty” state and can only be integrated into an empty data file. Consequently, a log backup named, for example, log[0025].4BL must be interpreted as the “26<sup>th</sup> backup of the log file, corresponding to operations performed between the 25<sup>th</sup> and 26<sup>th</sup> database backup”. log[0025].4BL thus corresponds to the backup[0025].4BK backup.

---

*Note* The backup numbering ranges are as follows:

- backup: 1 to 9999
  - log file backup: 0 to 9998
  - segment: 1 to 9999.
- 

**Last Backup Information** The “Last Backup Information” area provides information on the last database backup. Information is provided if at least one backup has taken place.

- **Last backup:** Date and time of the last backup.
- **Next scheduled backup:** Date and time of the next backup; information is provided in this area if a backup schedule has been put into place.
- **Backup file name:** Access path and file name of the last main backup. If the backup is segmented, the name of the first segment is displayed. You can click in the area indicating the location of the file in order to display its pathname as a pop-up menu containing the series of folders on the disk.
- **Log backup file name:** Access path and file name of the last log file backup (if any). You can click in the area indicating the location of the file in order to display its pathname as a pop-up menu containing the series of folders on the disk.
- **Status:** This area displays the error code of the last backup, as well as a description of this code. If the backup was executed properly, the area indicates “No detected error.”  
For scheduled backups, you can use this area to verify that the last backup occurred as scheduled.

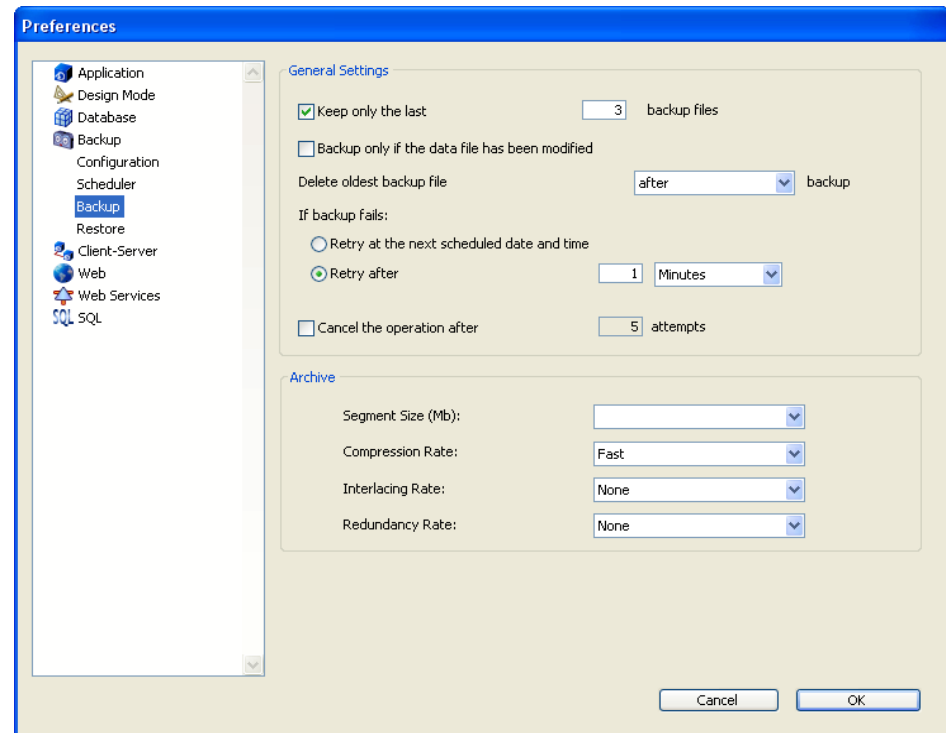
This information is also displayed on the Backup page of the MSC (see the “Backup” paragraph on page 1258).

*Note* The parameters related to log file management are covered in the “Managing the Log File” paragraph on page 1226.

## Backup Settings

Like the configuration settings, backup settings are used for each backup. Moreover, any changes to these settings are optional. Their default values correspond to the standard use of the backup function.

The backup settings are defined on the **Backup** page of the application Preferences:



*4D Server* The backup settings can only be set from the 4D Server machine.

These settings are specific to each database opened with the 4D application.

**Keep Only the Last X Backup Files**

This parameter allows activating and configuring the mechanism used to delete the oldest backup files. It lets you keep a specific number of the last backup files on the backup disk — the oldest file is then deleted at each new backup— and thus avoids the risk of saturating the disk drive.

This feature works as follows: once the current backup is complete, 4D deletes the oldest archive if it is found in the same location as the archive to back up and has the same name (you can request that the oldest archive be deleted before the backup in order to save space). If, for example, the number of sets is set to 3, the first three backups create the archives MyBase-0001, MyBase-0002, and MyBase-0003 respectively. During the fourth backup, the archive MyBase-0004 is created and MyBase-0001 is deleted.

Based on the space on the disk that you set aside for your backups, you can determine the maximum possible number of backup sets using the following equation:

$$\text{Maximum number of sets} = \frac{\text{Space available for backups}}{\text{Maximum backup size}} - 1 \text{ set}$$

You must reduce the maximum number of sets by 1 because 4D, by default, first performs the current backup, then deletes the oldest archive from the disk. This behavior can be changed (please see the [“Delete Oldest Backup File Before/After Backup” paragraph on page 1221](#)).

By default, the mechanism for deleting sets is enabled and 4D keeps 3 backup sets.

To disable the mechanism, simply deselect the option.

---

*Note* This parameter concerns both the database backup sets and the log file backup sets.

---

**Backup Only if the Data File has been Modified**

When this option is checked, 4D starts scheduled backups only if data has been added, changed or deleted in the database since the last backup. If not, the scheduled backup is cancelled and carried over until the next scheduled backup. No error is generated; however the backup journal notes that the backup has been postponed.

This option also allows saving machine time for the backup of databases mainly used for viewing purposes. Please note that enabling this option does not take into account any modifications made to the structure file or attached files.

- 
- Notes*
- For more information on scheduled backups, please refer to the [“Scheduled Backup Settings” paragraph on page 1223](#).
  - Manual backups of the database do not take this option into account.
  - This parameter concerns both database and log file backups.
- 

### Delete Oldest Backup File Before/After Backup

This option is only used if the “Keep only the last X backup files” option is checked. It allows you to specify whether 4D should start by deleting the oldest archive before starting the backup (**before** option) or if the deletion should occur once the backup is completed (**after** option). In order for this mechanism to work, the oldest archive must not have been renamed or moved.

The **before** option allows you to save space. It is not necessary to reduce the number of backups allowed by 1, which lets you, for example, store two 2-GB archives on a 5-GB disk. Please also note that in cases of interruption during a backup, you can access neither the old backup (since it was previously deleted), nor the current backup, which is not yet completed. Since the database could not be modified during the backup, you can be sure that your data file is intact, even if you do not have a valid backup. Once the system restarts and the incomplete archive is destroyed, proceed immediately with a new backup.

The **after** option is an additional security measure but requires more space on the backup hard disk.

### If Backup Fails

This option allows defining the mechanism used to handle failed backups (backup impossible).

- 
- Notes*
- Not all incidents result in a failed backup (please refer to the [“If Backup Fails” paragraph on page 1221](#)).
  - 4D considers a backup as failed if the database was not launched at the time when the scheduled automatic backup was set to be carried out (please refer to the [“Scheduled Backup Settings” paragraph on page 1223](#)).
-

When a backup cannot be performed, 4D allows making a new attempt. Two options are available:

- **Retry at the next scheduled date and time:** This option only makes sense when working with scheduled automatic backups (please refer to the [“Scheduled Backup Settings” paragraph on page 1223](#)). It cancels the failed backup and an error is generated.
- **Retry after X Seconds, Minutes or Hours:** When this option is checked, a new backup attempt is executed after the wait period. This mechanism allows anticipating certain circumstances that block the backup. You can set a wait period in seconds, minutes or hours using the corresponding menu. If the new attempt also fails, an error is generated and the failure is noted in the status area of the last backup and in the backup journal file.
- **Cancel the operation after X attempts:** This parameter is used to set the maximum number of failed backup attempts. If the backup has not been carried out successfully after the maximum number of attempts set has been reached, it is cancelled and the error 1401 is generated (“The maximum number of backup attempts has been reached; automatic backup is temporarily disabled”). In this case, no new automatic backup will be attempted as long as the application has not been restarted, or a manual backup has not been carried out successfully.

This parameter is useful in order to avoid a case where an extended problem (requiring human intervention) that prevented a backup from being carried out would have led to the application repeatedly attempting the backup to the detriment of its overall performance. By default, this parameter is not checked.

### Archive

This area allows setting archive generation options. These options apply to main backup files and to log file backup files.

- **Segment Size (MB):**

4D allows you to segment archives, which is to cut an archive up into smaller sizes. This behavior allows you, for example, to store a backup on several different disks (CDs, ZIPs, etc.). During a restore process, 4D will automatically fuse the segments. Each segment is called *MyDatabase[xxxx-yyyy].4BK*, where *xxxx* is the backup number and *yyyy* is the segment number. For example, the three segments of the sixth MyDatabase database backup will be called *MyDatabase[0006-0001].4BK*, *MyDatabase[0006-0002].4BK* and *MyDatabase[0006-0003].4BK*.



The **Segment Size** menu is a combo box that allows you to set the size in MB for each segment of the backup. You can choose one of the pre-set sizes or enter a specific size between 0 and 2048. If you pass 0, no segmentation occurs (this is the equivalent of passing **None**).

■ **Compression Rate**

By default, 4D compresses backups to help save disk space. However, the file compression phase can noticeably slow down backups when dealing with large volumes of data.

The **Compression Rate** option allows you to adjust file compression:

- **None:** No file compression is applied. The backup is faster but the archive files are considerably larger.
- **Fast (default):** This option is a compromise between backup speed and archive size.
- **Compact:** The maximum compression rate is applied to archives. The archive files take up the least amount of space possible on the disk, but the backup is noticeable slowed.

■ **Interlacing Rate and Redundancy Rate**

4D generates archives using specific algorithms that are based on optimization (interlacing) and security (redundancy) mechanisms. You can set these mechanisms according to your needs. For these two options, 4D provides a default rate **Medium**. The menus for these options also contain rates of **Low**, **High** and **None**.

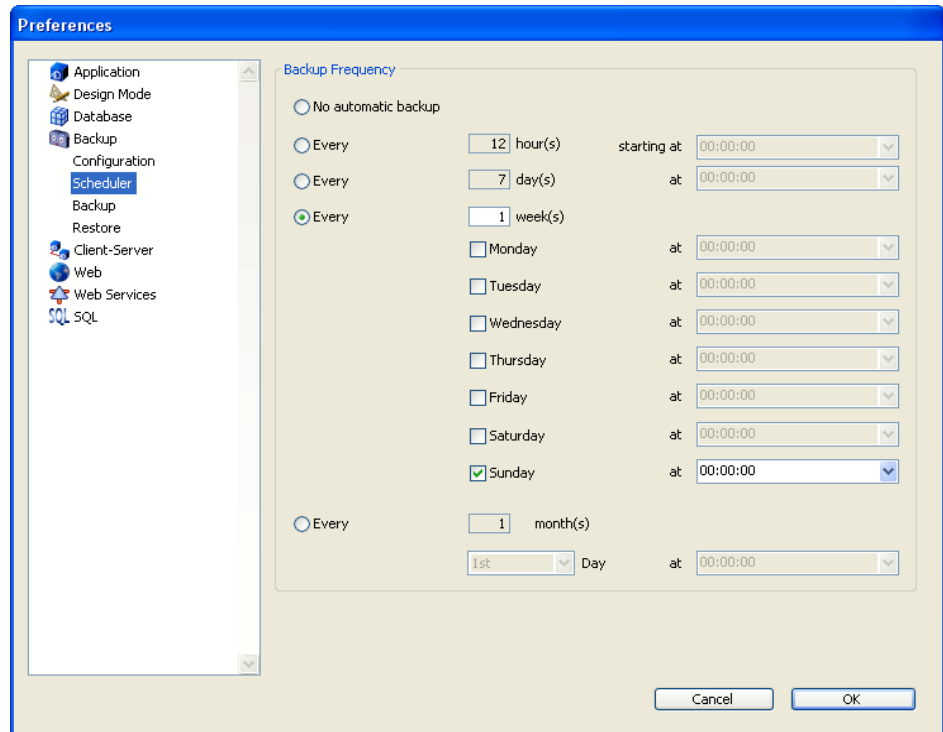
- **Interlacing Rate:** Interlacing consists of storing data in non-adjacent sectors in order to speed up their read time. However, the storage phase is slower.
- **Redundancy Rate:** Redundancy allows securing data present in a file by repeating the same information several times. The higher the redundancy rate, the better the file security; however, storage is slow and the file size is large.

## Scheduled Backup Settings

You can automate the backup of databases opened with 4D or 4D Server (even when no client machines are connected). This involves setting a backup frequency (in hours, days, weeks or months); for each session, 4D automatically starts a backup using the backup settings defined in the Preferences.

If this application was not launched at the theoretical moment of the backup, the next time 4D is launched, it considers the backup as having failed and applies the appropriate configuration, set in the database Preferences (refer to the “If Backup Fails” paragraph on page 1221).

The scheduler backup settings are defined on the **Scheduler** page of the application Preferences:



First select a frequency scale (days, weeks, etc.) by clicking on the corresponding radio button. You must then specify when to perform the backup.

- **No automatic backup:** The scheduled backup feature is disabled.
- **Every X hour(s):** This option allows programming backups on an hourly basis. You can enter a value between 1 and 24.
  - **starting at:** Sets the time at which the first hourly backup will begin.
- **Every X day(s) at x:** This option allows programming backups on a daily basis. Enter 1 if you want to perform a daily backup. When this option is checked, you must enter the time when the backup should start.

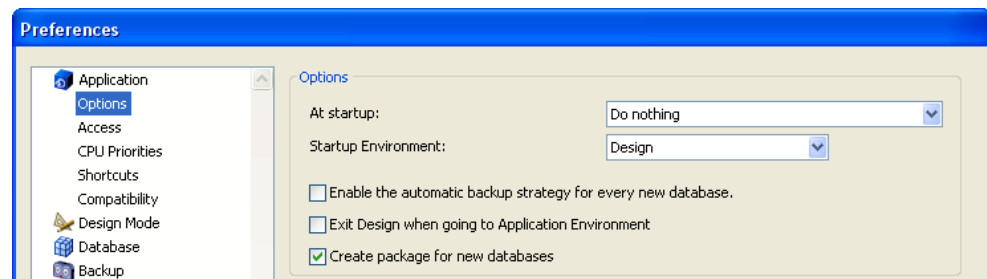
- **Every X week(s) day at x:** This option allows programming backups on a weekly basis. Enter 1 if you want to perform a weekly backup. When this option is checked, you must enter the day(s) of the week and the time when the backup should start. You can select several days of the week, if desired. For example, you can use this option to set two weekly backups: one on Wednesday and one on Friday.
- **Every X month(s), Xth Day at x:** This option allows programming backups on a monthly basis. Enter 1 if you want to perform a monthly backup. When this option is checked, you must indicate which day of the month and the time when the backup should start.

## Automatic Backup Strategy

4D provides a default backup strategy when a database is created. This strategy puts into place a set of backup parameters that assure minimum database security.

These settings can be changed once the database is created.

The backup strategy is enabled, by default, when a new database is created. To disable it, simply deselect the **Enable the automatic backup strategy for every new database** option in the Preferences dialog box (on the Application/Option page):



The automatic backup strategy includes the following settings:

- Creation and use of a log file (named *DataFileName.journal*).
- Backup of all database files (data, log, structure and user structure) next to the database structure file.
- Automatic backup every Sunday at 00:00:00.
- Default backup settings (keeping last 3 backups, medium rates, etc.).
- All the automatic restore options.

Of course, it is also possible to change these settings after the database has been created.

## Managing the Log File

A continuously-used database is always recording changes, and record additions or deletions. Performing regular backups of data is important but does not allow (in case of incident) restoring data entered since the last backup. To respond to this need, 4D now offers a specific tool: the log file. This file allows ensuring permanent security of database data.

In addition, 4D works continuously with a data cache in memory. Any changes made to the data of the database are stored temporarily in the cache before being written to the hard disk. This accelerates the operation of applications; in fact, accessing memory is faster than accessing the hard disk. If an incident occurs in the database before the data stored in the cache could be written to the disk, you must include the current log file in order to restore the database entirely.

### Introduction

The log file generated by 4D contains all operations performed on a database, which are logged sequentially. As such, each operation performed by a user causes two simultaneous actions: the first one in the database (instruction is executed normally) and the second one in the log file (the description of the operation is recorded). The log file is created independently without disturbing or slowing down the work of the user. A database can only work with one log file at a time.

The log file records the following types of operations:

- Opening and closing of the data file,
- Opening and closing of the process (contexts),
- Adding of records or BLOBs
- Modifying of records
- Deleting of records
- Creating and closing of transactions.

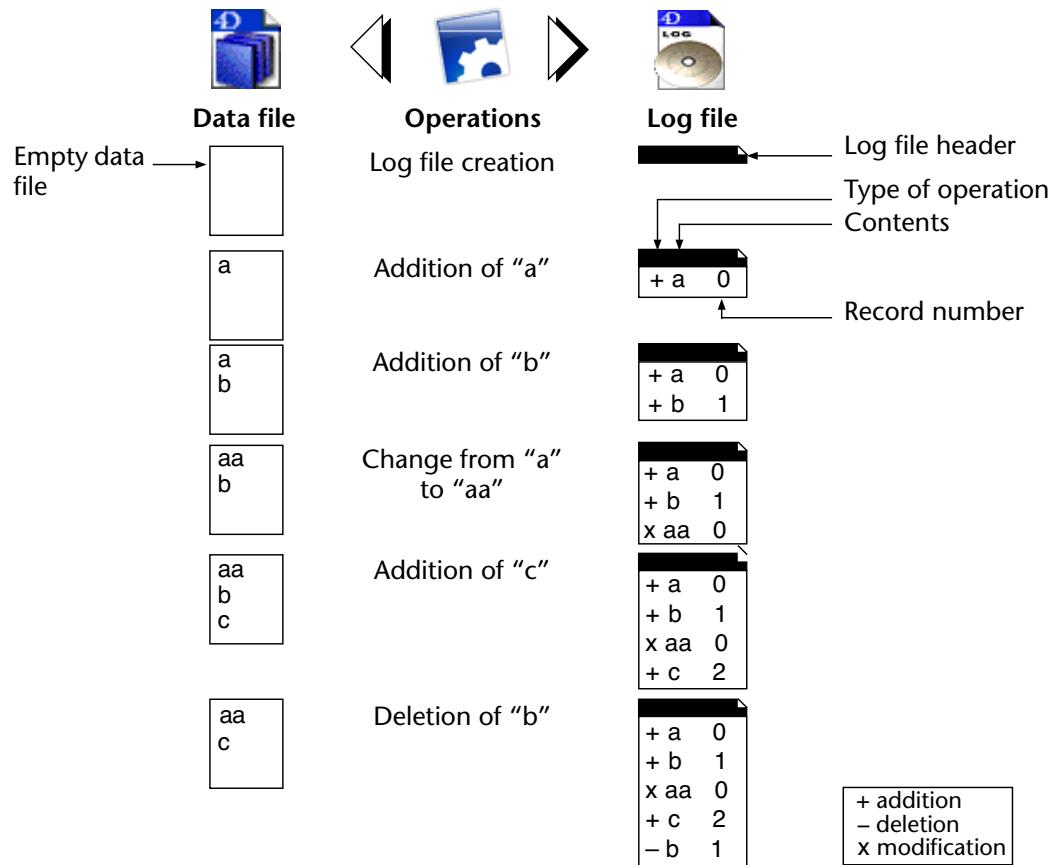
---

*Note* For more information about these actions, please refer to the [“Parsing a Log File” paragraph on page 1231](#).

---

4D manages the log file. It takes into account all operations that affect the data file equally, regardless of any manipulations performed by a user, 4D methods, the SQL engine, 4D plug-ins (4D Write, 4D View, etc.), or a Web browser.

The following illustration sums up how the log file works:



The current log file is automatically saved with the current data file. This mechanism has two distinct advantages:

- It avoids saturating the disk volume where the log file is stored. Without a backup, the log file would get bigger and bigger with use, and would eventually use all available disk space. For each data file backup, 4D or 4D Server closes the current log file and immediately starts a new, empty file, thereby avoiding the risk of saturation. The old log file is then archived and eventually destroyed depending on the mechanism for managing the backup sets.

- It keeps log files corresponding to backups in order to be able to parse or repair a database at a later point in time. The integration of a log file can only be done in the database to which it corresponds. It is important, in order to be able to properly integrate a log file into a backup, to have backups and log files archived simultaneously.

**Creating the Log File** By default, any database created with 4D uses a log file: the creation of this file is part of the automatic backup strategy (please refer to the [“Automatic Backup Strategy” paragraph on page 1225](#)). The log file is named **DataFileName.journal** and is placed in the folder containing the database structure.

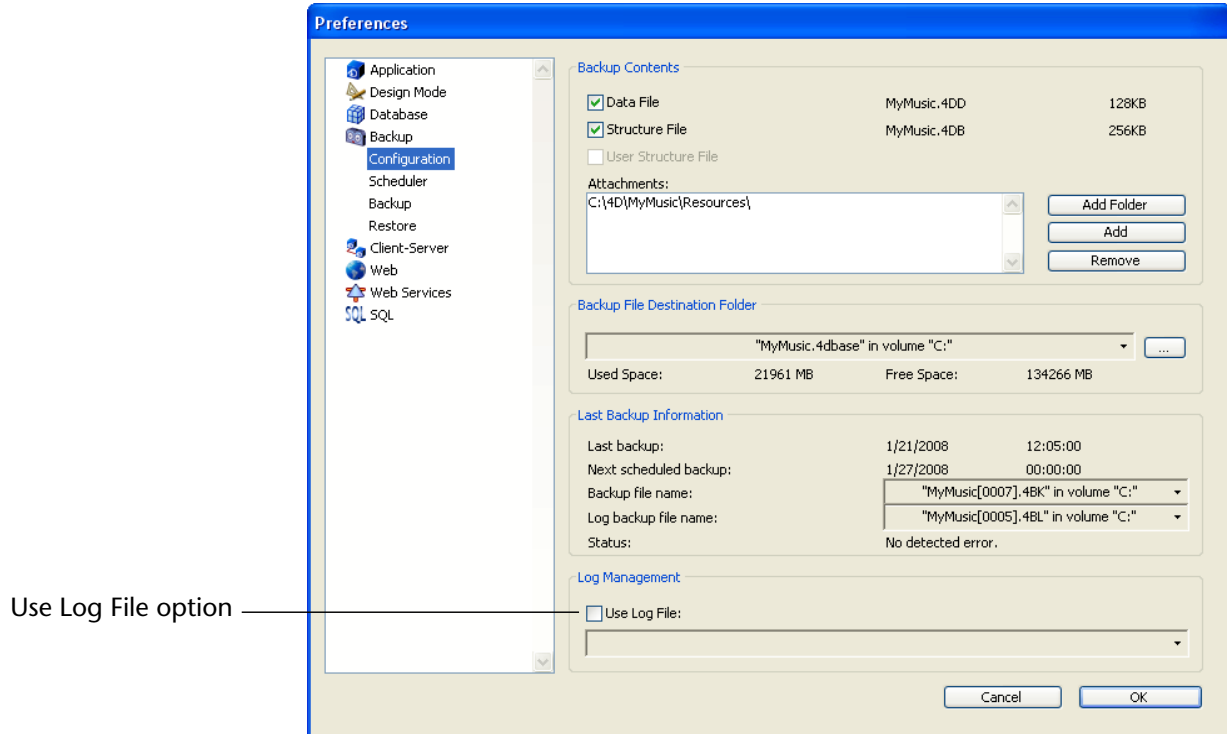
You can find out if your database uses a log file at any time: just check whether the **Use Log File** option is selected on the **Configuration** page of the Preferences (please refer to the [“Configuration of Backup Files” paragraph on page 1214](#)).



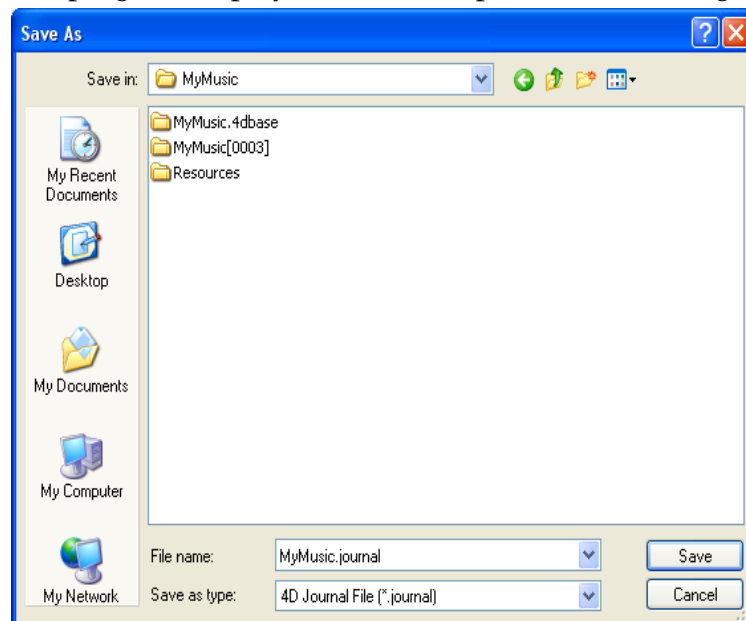
If you deselected this option, or if you use a database without a log file and wish to set up a backup strategy with a log file, you will have to create one.

► To create a log file:

- 1 On the **Configuration** page ("Backup" theme) of the application Preferences, check the **Use Log File** option.



The program displays a standard open file or new log file dialog box:



By default, the log file is named **DataFileName.journal**.

### 2 Keep the default name or rename it, and then select the file location.

If you have at least two hard drives, it is recommended that you place the log file on a disk other than the one containing the database. If the database hard drive is lost, you can still recall your log file.

### 3 Click Save.

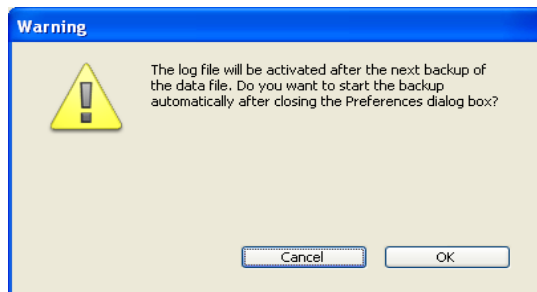
The disk and the name of the open log file are now displayed in the “Use Log File” area of the Preferences dialog box. You can click on this area in order to display a pop-up menu containing the series of folders on the disk.

### 4 Validate the Preferences dialog box.

In order for you to be able to create a log file directly, the database must be in one of the following situations:

- The data file is blank,
- You just performed a backup of the database and no changes have yet been made to the data.

In all other cases, when you validate the Preferences dialog box, an alert dialog box will appear to inform you that it is necessary to perform a backup.



If you click **OK**, the backup begins immediately, then the log file is activated. If you click **Cancel**, the request is saved but the creation of the log file is postponed and it will actually be created only after the next backup of the database.

This precaution is indispensable because, in order to restore a database after any incidents, you will need a copy of the database into which the operations recorded in the log file will be integrated.

Without having to do anything else, all operations performed on the data are logged in this file and it will be used in the future when the database is opened.

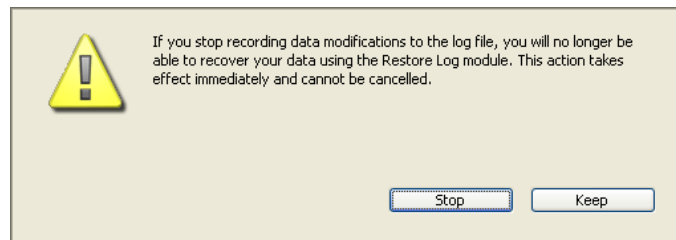


- You must create another log file if you create a new data file.
- You must set or create another log file if you open another data file that is not linked to a log file (or if the log file is missing).

## Stopping a Log File

If you would like to stop logging operations to the current log file, simply deselect the **Use Log File** option on the **Configuration** page (“Backup” theme) of the application Preferences.

4D then displays an alert message to remind you that this action prevents you from taking advantage of the security that the log file provides:



If you click **Stop**, the current log file is immediately closed (the Preferences dialog box does not need to be validated afterwards).

If you wish to close the current log file because it is too large, you must perform a data file backup, which will cause the log file to be backed up as well.

---

*4D Server* The New log file command can be used to automatically close the current log file and start a new one.

---

## Parsing a Log File

4D includes a function that allows viewing the contents of the current log file. This function is useful for parsing the use of a database or detecting the operation(s) that caused errors or malfunctions. In the case of a database in client-server mode, it allows verifying operations performed by each client machine.

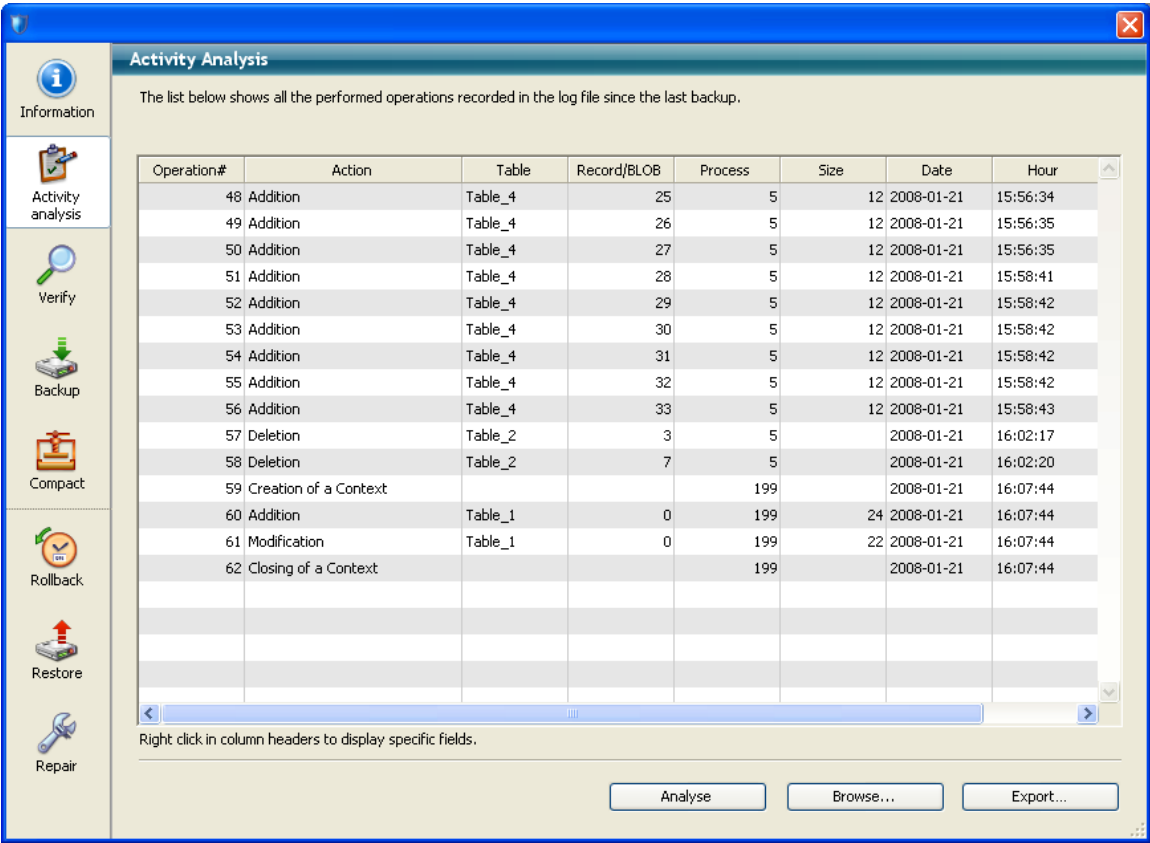
---

*Note* It is also possible to rollback the operations carried out on the data of the database. For more information about this point, please refer to the [“Undoing Operations” paragraph on page 1242](#).

---

The log file analysis and rollback functions are available in the Maintenance and Security Center (MSC).

To view the current log file, select **Maintenance Security Center** from the **Help** menu of 4D (or click on the MSC icon in the 4D tool bar) and display the “Activity analysis” page:



- Notes
- Only the Administrator and Designer of the database can access the log file verification function.
  - For more information about the MSC, please refer to the [chapter “Maintenance and Security Center”](#) on page 1247.

Every operation recorded in the log file appears as a row. The columns provide various information on the operation. You can modify the default column display — for example, you can display field values — using the contextual menu of the window (see the [“Setting the Column Display” paragraph on page 1234](#)). Moreover, the date and time when the data file was opened and/or closed is also displayed.

This information allows you to identify the source and context of each operation:

- **Operation#:** Sequence number of operation in the log file.
- **Action:** Type of operation performed on the data. This column can contain one of the following operations:
  - *Opening of Data File:* Opening of a data file.
  - *Closing of Data File:* Closing of an open data file.
  - *Creation of a Context:* Creation of a process that specifies an execution context.
  - *Closing of a Context:* Closing of process.
  - *Addition:* Creation and storage of a record.
  - *Adding a BLOB:* Storage of a BLOB in a BLOB field.
  - *Deletion:* Deletion of a record.
  - *Modification:* Modification of a record.
  - *Start of Transaction:* Transaction started.
  - *Validation of Transaction:* Transaction validated.
  - *Cancellation of Transaction:* Transaction cancelled.
- **Table:** Table to which the added/deleted/modified record or BLOB belongs.
- **Record/BLOB:** Record number or sequence number of the BLOB involved in the operation.
- **Process:** Internal number of process in which the operation was carried out. This internal number corresponds to the context of the operation.
- **Size:** Size (in bytes) of data processed by the operation.
- **Date and Hour:** Date and hour when the operation was performed.

- **User:** Name of the user that performed the operation. In client-server mode, the name of the client-side machine is displayed; in single-user mode, the ID of the user is displayed.  
If the 4D passwords are not enabled, this column is blank.
- **Values** (column not displayed by default): Values of fields for the record in the case of addition or modification. The values are separated by “;”. Only values represented in alphanumeric form are displayed.

Click on **Analyze** to update the contents of the current log file of the selected database (named by default *dataname.journal*).

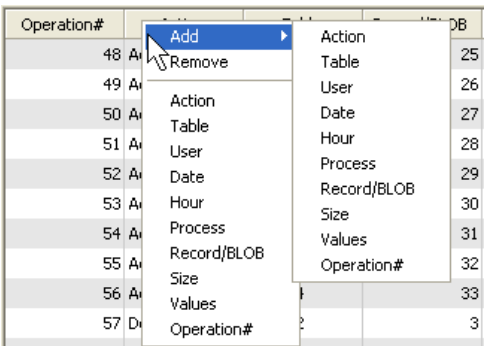
The **Browse...** button can be used to select and open another log file for the database.

The **Export...** button can be used to export the contents of the file as text.

Setting the Column Display

You can customize the column display using the contextual menu of the window. You can add or remove columns, or replace the contents of a column for the current window session (the customized window contents are reset when the session is closed).

To display the contextual menu of the window, click on a column header:



The following commands are available:

- **Add:** allows adding a column to the right of the column clicked on.
- **Remove:** allows removing the column clicked on.
- **Column Names:** allows replacing the contents of the column clicked on by another column of information.

## Restoring Data

The built-in 4D backup module allows you to restore entire sets of database data in case of any incidents, regardless of the cause of the incident.

### Incidents and Diagnostic

Two primary categories of incidents can occur:

- **The unplanned stoppage of a database while in use.**

This incident can occur because of a power outage, system element failure, etc. In this case, depending on the current state of the data cache at the moment of the incident, the restore of the database can require different operations:

- If the cache was empty, the database opens normally. Any changes made in the database were recorded. This case does not require any particular operation.
- If the cache contains operations, the data file is intact but it requires integrating the current log file.
- If the cache was in the process of being written, the data file is probably damaged. The last backup must be restored and the current log file must be integrated.

- **The loss of database file(s).**

This incident can occur because of defective sectors on the disk containing the database, a virus, manipulation error, etc.

The last backup must be restored and then the current log file must be integrated.

To find out if a database was damaged following an incident, simply relaunch the database using 4D. The program performs a self-check and details the necessary restore operations to perform.

In automatic mode, these operations are performed directly without any intervention on the part of the user (please refer to the following paragraph).

If a regular backup strategy was put into place, the 4D restore tools will allow you to recover (in most cases) the database in the exact state it was in before the incident. Two main tools are at your disposal:

- Backup restore — which consists in extracting files contained in one or more backup file(s) and making them readable again for the application that created them.

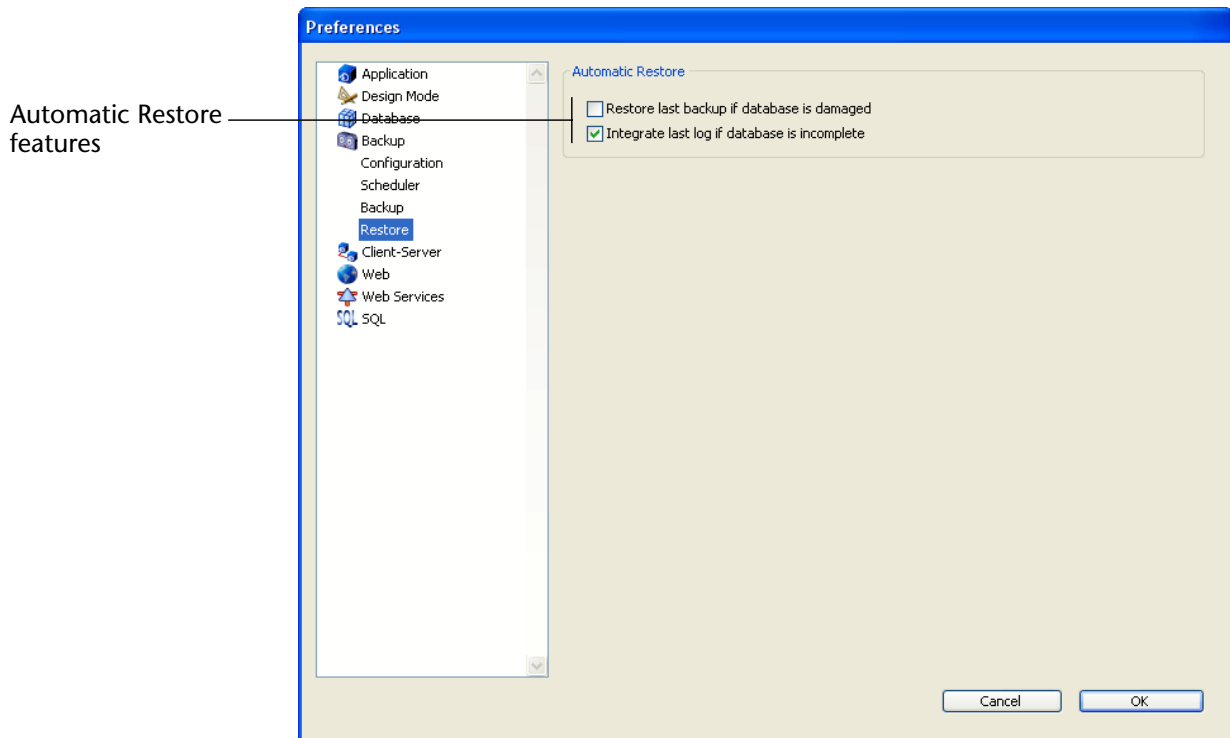
- Log file integration.

These two functions can be combined and automated.

**Automatic Restore** By default, 4D automatically launches the database restore procedures after an incident. Two types of automatic features are used:

- Automatic restore
- Automatic log file integration

These mechanisms can be disabled using the options available on the **Restore** page (“Backup” theme) in the application Preferences:



### Restore Last Backup if Database is Damaged

When this option is checked, the program automatically starts the restore of the data file of the last valid backup of the database, if an anomaly is detected during database launch. No intervention is required on the part of the user; the operation is logged in the backup journal (please refer to the [“Backup Journal” paragraph on page 1244](#)).

*Note* In the case of an automatic restore, only the data file is restored. If you wish to recover the attached files or the structure file, you must perform a manual restore.

---

The steps of the automatic restore are as follows:

- First, 4D renames the damaged data file.
- 4D extracts the data file of the last backup and stores it in place of the previous one.
- If the “Integrate last log if database is incomplete” option is checked, 4D integrates the log file, if necessary (please refer to the [“Integrate Last Log File if Database is Incomplete” paragraph on page 1237](#)).
- 4D opens the restored database.

#### **Integrate Last Log File if Database is Incomplete**

When this option is checked, the program automatically integrates the log file when a restored database is opened. No intervention on the part of the user is required.

- When opening a database, the current log file is automatically integrated if 4D detects that the operations stored in the log file are not present in the data. This situation occurs, for example, when a power outage occurs when there are operations in the data cache that have not yet been written to the log.
- When restoring a database, if the current log file or backup log file having the same number as the backup file is stored in the same folder, 4D examines its contents. If it contains operations not found in the data file, the program automatically integrates it.

The user does not see any dialog box; the operation is completely automatic. The goal is to make use as easy as possible. The operation is logged in the backup journal (please refer to the [“Backup Journal” paragraph on page 1244](#)).

#### **Manually Restoring a Backup (Standard Dialog)**

You can restore the contents of an archive generated by the backup module manually. A manual restore may be necessary, for instance, in order to reproduce the contents of an archive in full (structure files and/or enclosed attached files), or for the purpose of carrying out searches among the archives.

The manual restore can also be performed along with the integration of the current log file.

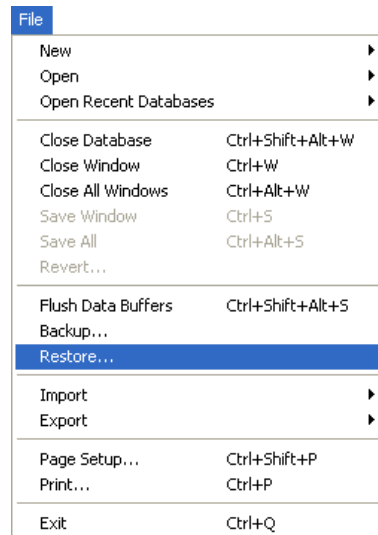
---

*Note* You can set 4D so that the database data are automatically restored in case of any incidents. For more information on this, please refer to the [“Automatic Restore” paragraph on page 1236](#).

---

The manual restore of backups can be carried out either via the standard Open document dialog box, or via the “Restore” page of the Maintenance and Security Center (MSC).

- Restoring via a standard dialog box can be used to restore any archive. This function is described below.
  - Restoring via the MSC provides more options and allows the archive contents to be previewed. On the other hand, only archives associated with the open database can be restored. This function is described in the [“Manually Restoring a Backup \(MSC\)”](#) paragraph on page 1240.
- To restore a database manually via a standard dialog box:
- 1 **Start the 4D application and choose Restore... in the File menu.**



It is not mandatory that a database be open.

**OR**

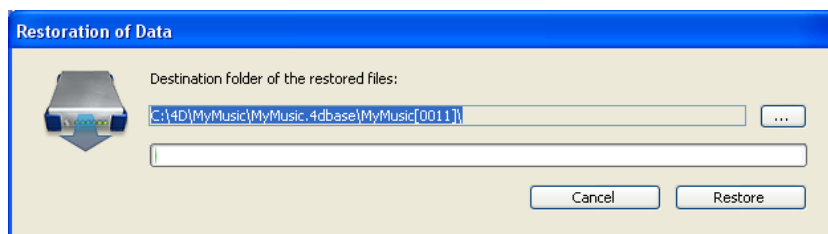
**Execute the RESTORE command from a 4D method.**

A standard Open file dialog box appears so that you can indicate the backup file (.4bk) or the backup log file (.4bl) to be restored.

- 2 **Select the file to restore and click Open.**



The following dialog box appears, which allows you to specify the location where files will be restored:



By default, 4D restores the files in a folder named “Archivename” (no extension) located next to the archive. You can click on the [...] button to specify a different location.

**3 Click on the Restore button.**

4D extracts all backup files from the specified location.

If the current log file or a backup log file with the same number as the backup file is stored in the same folder, 4D examines its contents. If it contains operations not present in the data file, the program asks you if you want to integrate these operations.

---

*Note* Integration is done automatically if the “Integrate last log file...” option is checked (please refer to the [“Integrate Last Log File if Database is Incomplete”](#) paragraph on page 1237).

---

**4 (Optional) Click OK to integrate the log file into the restored database.**

If the restore and integration were carried out correctly, 4D displays a dialog box indicating that the operation was successful.

**5 Click OK .**

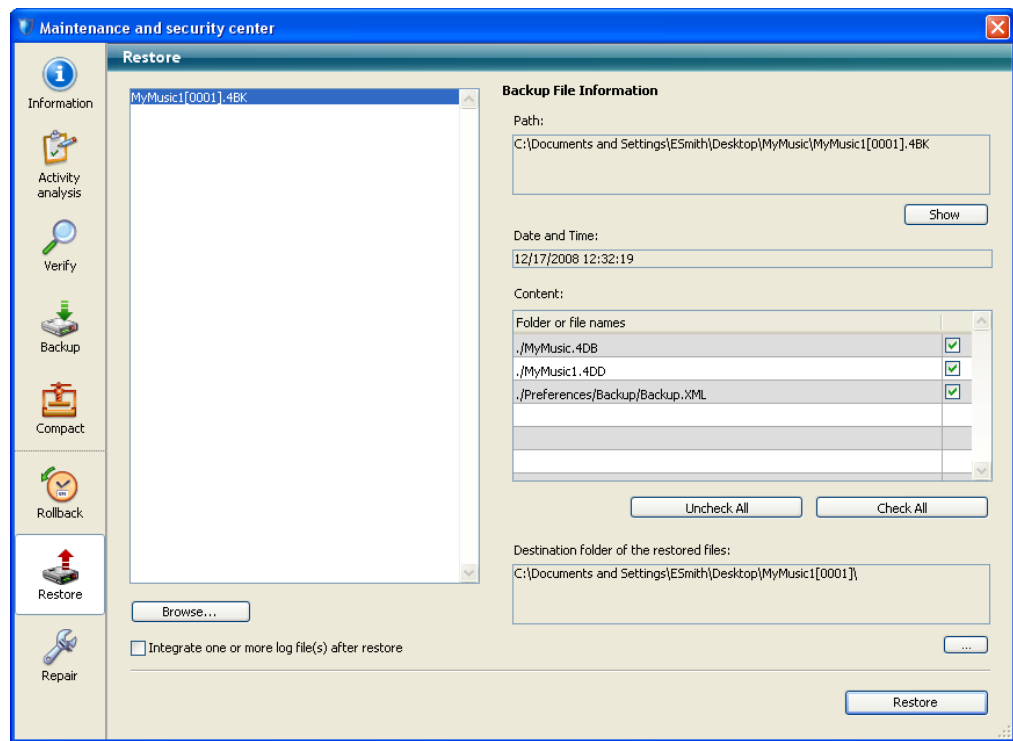
The destination folder is displayed. During the restore, 4D places all backup files in this folder, regardless of the position of the original files on the disk when the backup starts. This way your files will be easier to find.

### Manually Restoring a Backup (MSC)

You can manually restore an archive of the current database using the “Restore” page of the Maintenance and Security Center (MSC). This page provides several options that can be used to control the restore.

*Note* It is also possible to manually restore any archive via a standard Open document dialog ox (see the [“Manually Restoring a Backup \(Standard Dialog\)”](#) paragraph on page 1237).

To restore a backup of the current database, select **Maintenance Security Center** in the **Help** menu of 4D (or click on the MSC icon in the 4D tool bar) and display the “Restore” page:



*Note* For more information about the MSC, please refer to the [chapter “Maintenance and Security Center”](#) on page 1247.

The list found in the left part of the window displays any existing backups of the database.

You can also click on the **Browse...** button found just under the area in order to open any other archive file from a different location. It is then added to the list of archives

When you select a backup in this list, the right part of the window displays the information concerning this particular backup:

- **Path:** Complete pathname of the selected backup file. Clicking the **Show** button opens the backup file in a system window.
- **Date and Time:** Date and time of backup.
- **Content:** Contents of the backup file. Each item in the list has a check box next to it which can be used to indicate whether or not you want to restore it. You can also use the **Check All** or **Uncheck All** buttons to set the list of items to be restored.
- **Destination folder of the restored files:** Folder where the restored files will be placed. By default, 4D restores the files in a folder named "*Archivename*" (no extension) that is placed next to the database structure file. To change this location, click on [...] and specify the folder where you want the restored files to be placed.

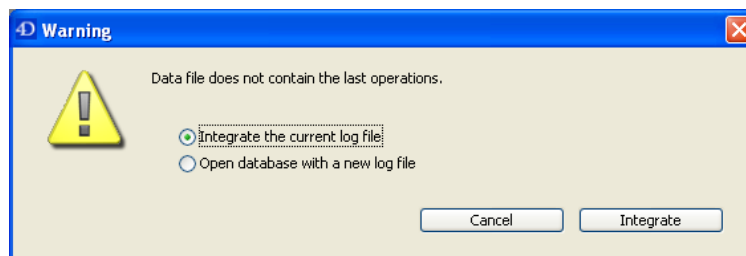
### Successive Integration of Several Log Files

The **Integrate one or more log file(s) after restore** option allows you to integrate several log files successively into a database. If, for example, you have 4 log archives (corresponding to 4 database backups), you can restore the first backup then integrate the log archives one by one. This means that you can, for example, recover a data file even when the last backup files are missing.

When this option is checked, 4D displays the standard Open file dialog box after the restore, which can be used to select log file to be integrated. The Open file dialog box is displayed again after each integration until it is cancelled.

### Manually Integrating the Log

If you have not checked the option for the automatic integration of the log file (see the "[Integrate Last Log File if Database is Incomplete](#)" paragraph on page 1237), a warning dialog box appears during the opening of the database when 4D notices that the log file contains more operations than have been carried out in the database.



In order for this mechanism to work, 4D must be able to access the log file in its current location.

You can choose whether or not to integrate the current log file. Not integrating the current log file allows you to avoid reproducing errors made in the data.

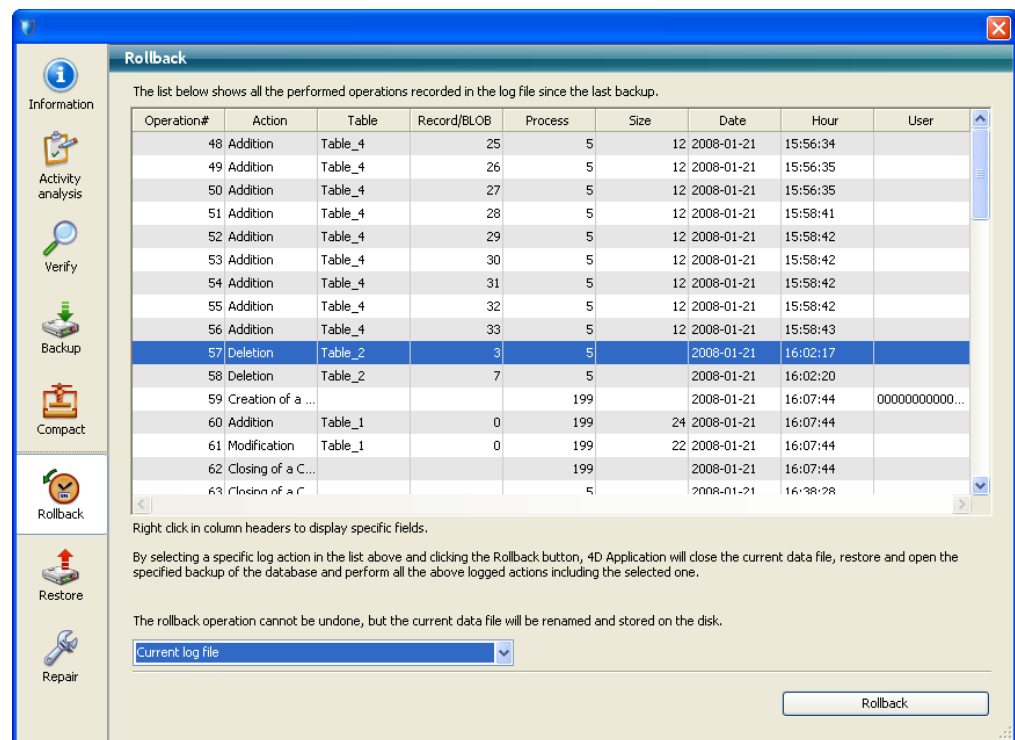
**Undoing Operations** 4D features a function that allows undoing operations performed on the log file. This function is similar to a multi-level cancel. It is especially useful when a record was deleted by error from a database.

In order for this function to be available, the database must have a log file.

- To restore a database to a previous state:

**1 Select Maintenance Security Center in the Help menu of 4D and display the “Rollback” page.**

The “Rollback” window appears:



**Note** Only the Administrator and Designer of the database have access to check the log file.

The list of operations is identical to that of the Activity analysis window. For more information about this list, please refer to the [“Parsing a Log File” paragraph on page 1231](#)).

## 2 Select the row after which all operations must be cancelled.

The operation of the selected row will be the last kept. If, for example, you wish to cancel a deletion, select the operation located just before it. The deletion operation, as well as all subsequent operations, will be cancelled:

Operation#	Action	Table	Record/BLOB	Process	Size	Date	Hour	User
48	Addition	Table_4	25	5	12	2008-01-21	15:56:34	
49	Addition	Table_4	26	5	12	2008-01-21	15:56:35	
50	Addition	Table_4	27	5	12	2008-01-21	15:56:35	
51	Addition	Table_4	28	5	12	2008-01-21	15:58:41	
52	Addition	Table_4	29	5	12	2008-01-21	15:58:42	
53	Addition	Table_4	30	5	12	2008-01-21	15:58:42	
54	Addition	Table_4	31	5	12	2008-01-21	15:58:42	
55	Addition	Table_4	32	5	12	2008-01-21	15:58:42	
56	Addition	Table_4	33	5	12	2008-01-21	15:58:43	
57	Deletion	Table_2	3	5		2008-01-21	16:02:17	
58	Deletion	Table_2	7	5		2008-01-21	16:02:20	
59	Creation of a ...			199		2008-01-21	16:07:44	000000000000...
60	Addition	Table_1	0	199	24	2008-01-21	16:07:44	
61	Modification	Table_1	0	199	22	2008-01-21	16:07:44	
62	Closing of a C...			199		2008-01-21	16:07:44	
63	Closing of a C...			5		2008-01-21	16:08:28	

## 3 Click **Roll back**.

4D asks you to confirm the operation.

## 4 Click **OK**.

The data is then restored to the exact state it was in at the moment of the selected action.

The menu found at the bottom of the window can be used to select a log file to be used when you apply the rollback function to a restored database. In this case, you must specify the log file corresponding to the archive.

## How it Works

When the user clicks the **Roll back** button, 4D shuts the current database and restores the last backup of the database data. The restored database is then opened and 4D integrates the operations of the log file up through to the selected operation.

If the database has not yet been saved, 4D starts with a blank data file.

## Backup Journal

To make following up and verifying database backups easier, the backup module writes a summary of each operation performed in a special file, which is similar to an activity journal. Like an on-board manual, all database operations (backups, restores, log file integrations) are logged in this file whether they were scheduled or performed manually. The date and time that these operations occurred are also noted in the journal.

The backup journal is named “Backup Journal.txt” and is placed next to the database data file.

The backup journal can be opened with any text editor. To make parsing easier, information in the journal is separated with tabs and each line of information ends with a carriage return.

### Management of Backup Journal Size

In certain backup strategies (for example, in the case where numerous attached files are being backed up), the backup journal can quickly grow to a large size.

Two mechanisms can be used to control this size:

- Automatic backup,
- Possibility of reducing the amount of information recorded.

### Automatic Backup

An automatic mechanism can be used to limit the size of the backup journal: before each backup, the application examines the size of the current backup journal file. If it is greater than 10 MB, the current file is archived and a new file is created. The archived files are renamed “Backup Journal[xxx].txt”, where xxx is a number from 1 to 999. Once file number 999 is reached, the numbering begins at 1 again and the existing files will be replaced.

### Reduce Amount of Information Stored

It is possible to reduce the amount of information recorded in the backup journal. To do this, simply modify the value of the `VerboseMode` key in the *Backup.xml* file of the database. By default, this key is set to `True`.

If you change the value of this key to `False`, only the main information will be stored in the backup journal: date and time of start of operation and any errors encountered.

The XML keys concerning backup configuration are described in the *XML Keys-Backup* manual.

## Using a backup.xml File

The backup preferences of 4D are available as an XML file associated with the database. 4D uses the data of this file to set the backup options (options found in the Preferences dialog box) when each backup is launched. This file can also be used to read or set additional options, such as the amount of information stored in the backup journal (see previous paragraph).

The XML backup configuration file is named **backup.xml**. It is automatically created at the following location:

**DatabaseFolder/Preferences/Backup/Backup.xml**

... where *DatabaseFolder* is the folder containing the database structure file. The Preferences/Backup subfolders are automatically created when necessary.

The XML keys concerning backup configuration are described in the *XML Keys-Backup* manual.

