4D 2004 Integrated Backup Module

By Yvan Ayaay, Technical Support Engineer, 4D, Inc.
TN 05-46

Introduction

The backup and restore operations have now been integrated in 4D 2004 (4th Dimension and 4D Server). These operations can be performed manually, automatically, or programmatically. The new features that come with the integrated backup module simplify the copying of the database together with its necessary files and restoring it back to its previous state. In this technical note, the basic features of 4D 2004 integrated module will be discussed. Among that will be covered are its basic configuration and ways to execute backup and restore, including the use of the log file.

Overview

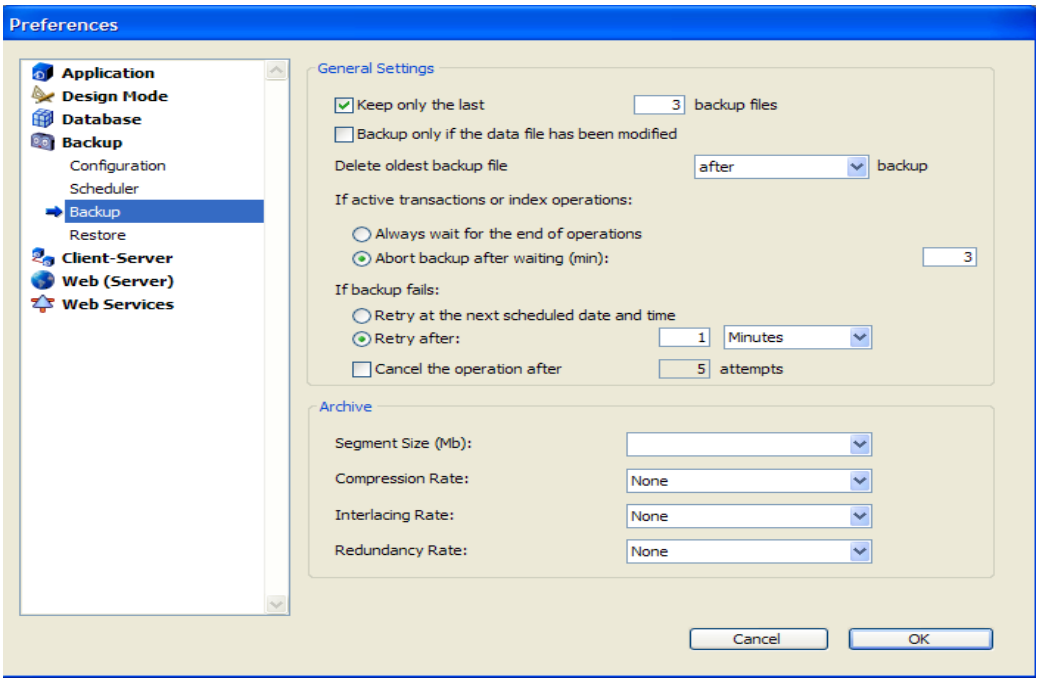
In the previous versions of 4D, the backup and restore operations are handled using the 4D Backup plug-in. In 4D 2004, similar operations are now integrated in the application. You can either start a backup manually, automatically, or programmatically from within 4D. You can make a copy of the database at anytime manually or use the backup scheduler to do it in time intervals. New simplified backup commands can be used to perform and manage the backup programmatically. The backup and restore parameters are configured in the database preferences. Each backup created can include the structure file and the data file together with additional files and folders. And if a log file is used, a log file backup is also saved. The backup can be restored to the previous state of the database in the event that something happens to the current database using the log file.

Backup Settings and Configuration

The backup settings and the configuration settings are used each time a backup is performed. In this section, a brief discussion of the backup settings and configuration settings will be conducted.

Backup Settings

The backup settings are set in the backup page under the Backup theme of the application preferences. The default values of backup, as shown below, are set to the standard use of the backup function. Changing these values is optional. The backup settings are defined in two sections: General Settings and Archive.



General Settings

Under the General Settings, you can define the number of backups to keep and also just perform backup if there is a modification with the data file. These options allow you to save space on your hard drive. The "Keep only the last x backup file" option, for instance, allows you to keep only number of x backups. That is, when a new backup (x +1) is created, the oldest backup file is deleted. You can set when to delete the backup either after or before creation of the backup by switching the "Delete oldest backup file after/before backup" option. When you're performing the backup automatically using the scheduler, you can also set "Backup only if data file has been modified" so that a scheduled backup is cancelled when no modification to the data file is done. As a result, the backup can be later performed on the next schedule.

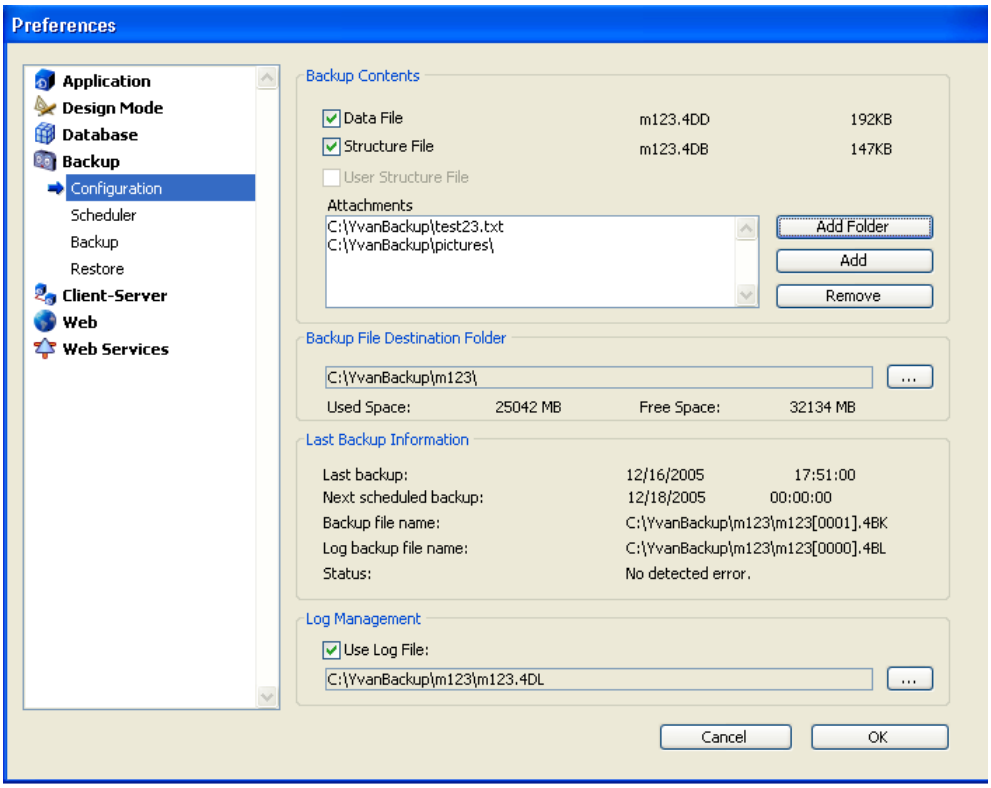
In addition, you can set the operation to be performed when there is an active transaction operation in the database and what to do when the backup fails.
You can either wait for a transaction to complete before starting a backup or wait for a set time before canceling the backup process. When the backup is cancelled, the status of the backup is considered failed. Thus, you can then implement mechanisms when a failed backup occurs. These options are either to retry at the next scheduled date and time or retry after a set time. You can choose to cancel the operation after certain attempts to do this.

Archive

Under the Archive portion, you can define archive generating options such as segmenting the archive, compressing the archive, and using optimizing methods. The archive can be segmented and saved on different locations. Furthermore, you can compress the archive to save disk space though this can slow down the backup process. Among the optimization options, you can set the rate of interlacing and redundancy.

Configuration Settings

When you perform the backup, the current backup configuration is used as well. This can be set in the Configuration settings of backup in the database preferences as shown below. It is divided into four sections: Backup Contents, Backup File Destination Folder, Last Backup Information, and Log File Management.



Backup Contents

Under the Backup Contents, you specify what to include in the backup file. The Structure file and Data file are selected by default. You can add files and folders to the backup file. As shown above, the file test23.txt and the folder pictures are attached.

Backup File Destination Folder

The destination of the backup is displayed in the Backup File Destination Folder field. You can choose a different destination location using the browse button (…).

Last Backup Information

You can get information about the last backup such as when it was done, the next scheduled backup, the name of the last backup file, log file backup, and the last backup status under the Last Backup Information section.

Log Management

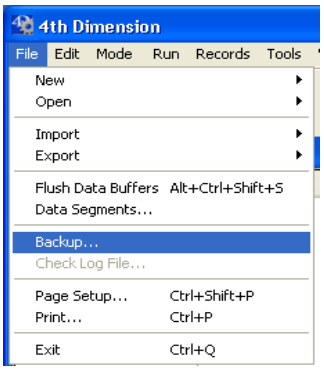
Under the Log Management, you can choose to use a log file. The log file contains changes made to the data after a full backup. Operations such as addition, modification, and deletion of records are recorded. Transactions are recorded in the log file as well.

Ways to perform backup

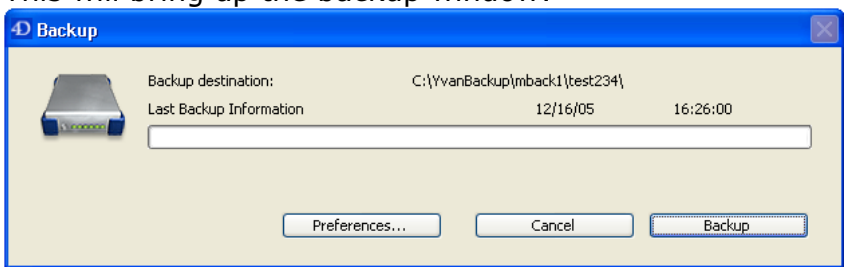
In 4D 2004, backup can be started manually, automatically, or programmatically.

Manually

You can create a backup manually at anytime by clicking on the Backup item of the File menu of 4D Server or of the File menu in the user environment of 4th Dimension as shown below:



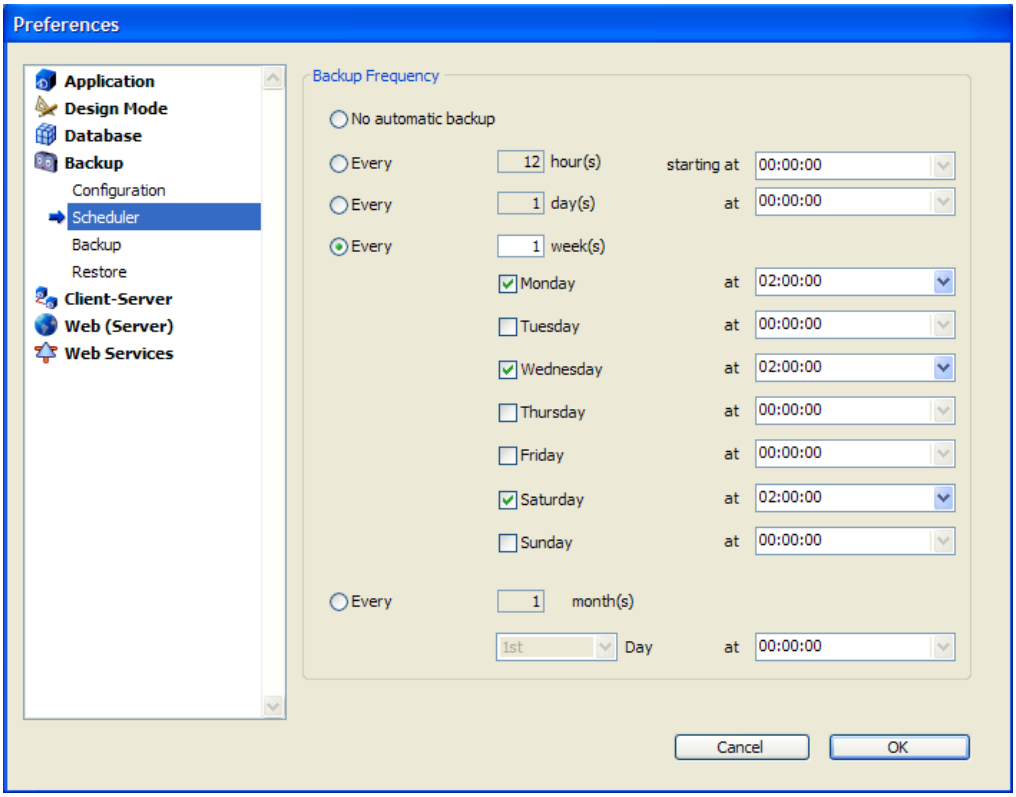
This will bring up the backup window:



Clicking on the Backup button starts the backup process which should use the current settings in the backup preferences. The button Preferences opens up the backup database preferences which could be changed before starting the backup.

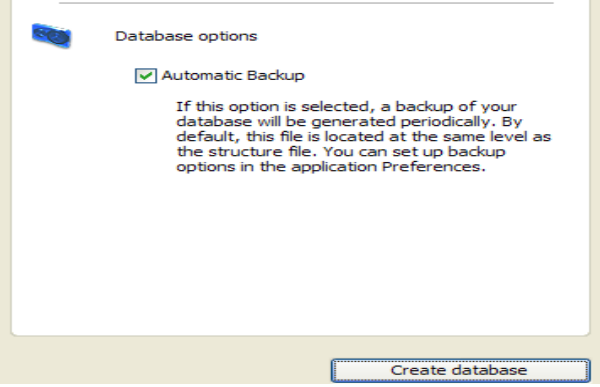
Automatically

You can automatically perform backup without user intervention by setting the frequency of the backup in the Scheduler page of the Backup preferences as shown below.



You can set the backup frequency every hour, every day, every week at certain days, or every month. To disable this scheduled backup feature, the "No automatic backup" option should be selected. Backup will be performed on the set time and date of the scheduler. As shown in the above settings, the backup will be done every 2 am of Mondays, Wednesdays, and Saturdays of every week. In the General Settings of the backup settings as discussed in the above section, you can choose not to perform a backup when no modification is done to the data file and perform the backup on the next scheduled backup instead.

When a database is first created, the automatic backup is selected by default as shown below. This uses default backup settings. The said option should be unchecked before creating a database to disable this feature.



Programmatically

You can programmatically perform a backup by using the *Backup* command. Along with the new database methods *On Backup Startup Database Method* and *On Backup Shutdown Database Method*, you can control the backup process. When the Backup command is executed, it uses the current backup settings and configuration. Just like the way the methods are named, the *On Backup Startup Database Method* is executed when the backup process is started and the *On Backup Shutdown Database Method* is executed when the backup process ends.

The new backup database methods allow you to set preconditions when performing a backup. For instance, you can just allow certain users to perform the backup, or you can choose to cancel the backup when a backup was already done on the same day, or even so, you can check the destination where the backup will be saved and alert the user when it will be saved on the same volume. All these can be done in the *On Backup Startup Database method*. The value returned for $0 in this method determines to authorize or refuse the backup. To authorize, $0 must be set to zero. Any other value will cancel the backup. The value passed in $0 can be checked in the *On Backup Shutdown Database Method*. This way, you can customize an error handling based on the value that is passed.

Whenever the backup ends, the *On Backup Shutdown Database Method* is executed. You can check the status of the backup using the *Get Backup Information* command. Or you can perform certain actions based on the value returned by the *On Backup Startup Database Method*. This value is passed into $1 integer variable. This variable should be declared in the method.

Below is an example on how to manage backup programmatically using the new backup commands. Conditions as shown below in the code can be checked in the *On Backup Startup Database Method* when the backup is executed with the *Backup* command. You can pass a value other than 0 to the $0 parameter to designate each condition not being met. These values can be checked in the On Backup Shutdown Database Method.

```
`Database Method: On Backup Startup Database Method.
`Conditions be met to execute the backup:
 `-Only the Administrator can perform the backup.
` -Backup will not be performed when a backup has already been made the same day.
` -Destination of the backup is checked. If the destination is on the same volume as the current
`   location of the database, the user is notified. The user can then choose to proceed or to cancel
`   the backup.

C_INTEGER(selector;$0)
C_TEXT(Vol_Structure;Vol_BackupDestination;dest_path)
selector:=0
$0:=0

If (Current user="Administrator") `allow  backup only if the user is Administrator.
  GET BACKUP INFORMATION(0;last_backupDate;last_backupTime)
     If (last_BackupDate=Current date) `only  perform backup on a different day
        $0:=3 `backup has already been done on the same day.
     Else
        Vol_Structure:=ExtractVolume (Structure file)
        `Get the current volume of the structure file.
        Vol_BackupDestination:=ExtractVolume (CheckDestination )
        `Get volume where backup will be saved.

           if (Vol_Structure=Vol_BackupDestination)
        ` if backup destination is on the same volume as the current database
              CONFIRM("Are you sure you want to perform backup on the same volume?";"Yes";"No")
   ` confirm from user to perform backup
              If (OK#1)
              `If user does not want to save backup on the same volume as the database.
               $0:=2
               ` client does not want to perform backup on the same volume
              End if
           End if
     End if
Else
   $0:=1 ` user is not administrator
End if
```

The value of the integer variable $0 in the *On Backup Startup Database Method* determines whether to allow or refuse backup. Backup is authorized if $0 is zero, otherwise, the backup is refused. In the above method, the current user is first checked. If the current user is the Administrator, the current date is then checked with the last backup date. If the backup is not performed on the same day, the backup destination volume is next checked with the current location of the structure file. If the location is on a different volume, the backup is performed. Otherwise, a message is displayed alerting the user and prompts the user to confirm. The value assigned to $0 depends whether the conditions are met.

The value returned in the *On Backup Startup Database Method* can be checked in the On Backup Shutdown Database Method. The integer variable $1 will contain the value returned by the *On Backup Startup Database Method*. You can customize error

handling using this value. You can use the *GET BACKUP INFORMATION* command to check on the backup error that occurred. As shown in the code below, if any one of the preconditions of starting the backup specified above is not met, the backup is not performed and the reason why it was not performed is displayed. If the error is not related to the conditions not being met, the error number and description is displayed.

```
 `Method:  On Backup Shutdown Database Method

C_INTEGER(error_num;$1)
C_TEXT(error_desc)

Case of
   : ($1=1)
      ALERT("Backup  was not performed because the current user is not the administrator. ")
   : ($1=2)
      ALERT("Backup  was not performed because you prefer not to save the backup on the same vo"+"lume
as your database.")
    : ($1=3)
      ALERT("Backup  was not performed because a backup was just performed today. ")
    : ($1=0)
      ALERT("Backup  Successful.")
   Else  `  An error occurred during backup.
      GET BACKUP INFORMATION(2;error_num;error_desc)
      ALERT("Error:  "+String(error_num)+"    - "+error_desc)
End case
```

The backup settings and configuration are saved in an XML file (backup.xml). The information displayed in the backup database preferences are mostly taken from this XML file. You can then programmatically set and retrieve backup settings using 4D XML commands. In the On Backup Startup Database Method above, a method called CheckDestination gets the backup destination folder from the backup.xml file. The code for this method is displayed below. In the method, the backup.xml is first parsed. Then, the destination folder element is located and its value retrieved and returned by the method.

```
 `Method:  CheckDestination
 ` Description:  It will check and return the backup destination folder.
C_TEXT(Path_Structure;$0)
C_TEXT(Path_Prefs;Path_Structure;Path_BackupXML)
C_TEXT($elementRef1;$elementRef2)
C_INTEGER($separator;$StrPtr)
C_INTEGER($myPlatform;$mySystem;$myMachine)

PLATFORM PROPERTIES($myPlatform;$mySystem;$myMachine)
If ($myPlatform=Windows )  `  check if platform is Windows or Mac
   $separator:=Ascii("\\")
Else
   $separator:=Ascii(":")
End if

Path_Structure:=Structure file  `  get the pathname of Structure file.
```

```
$StrPtr:=Length(Path_Structure) ` get length of string
While (Path_Structure[[$StrPtr]]#Char($separator)) ` starting from the end of string, find first occurence of the separator
    $StrPtr:=$StrPtr-1
End while
$var1:=Substring(Path_Structure;1;$StrPtr) ` extract path to preferences
Path_Prefs:=$var1+"Preferences"
Path_BackupXML:=Path_Prefs+Char($separator)+"Backup"+Char($separator)+"Backup.xml" ` path to backup.xml file
$ref1:=DOM Parse XML source(Path_BackupXML)
$elementRef1:=DOM Find XML element($ref1;"/Preferences4D/Backup/Settings/General")
$elementRef2:=DOM Find XML element($elementRef1;"/General/DestinationFolder") ` find Destination folder element
DOM GET XML ELEMENT VALUE($elementRef2;value) ` get element value for destination folder
DOM CLOSE XML($ref1)

$0:=value ` return destination folder of backup.
```

Note: In 4D Server, when a backup command is executed on the client, the backup is considered as a stored procedure and is executed on the server.

Using a Log File

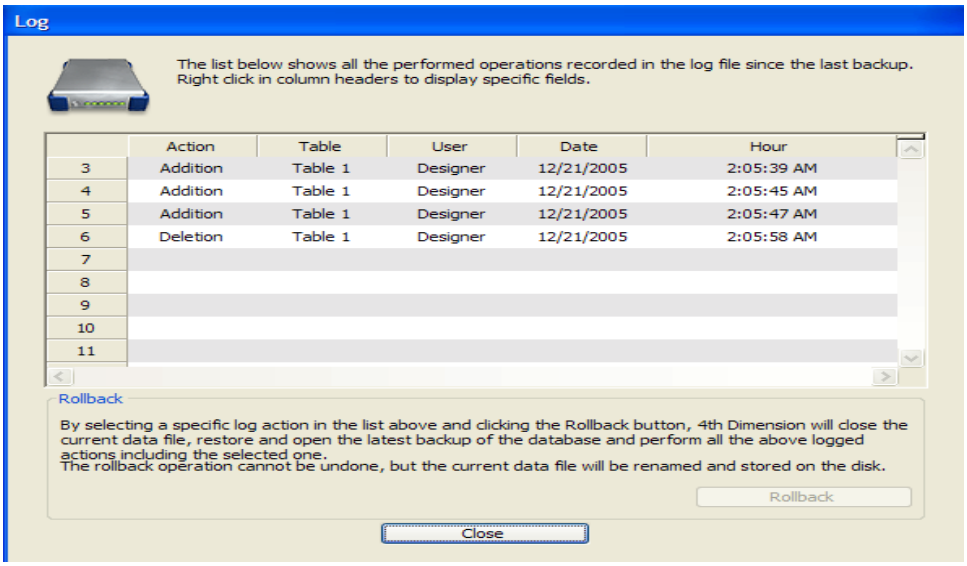
Restoring a previous backup alone may not be enough to restore the database to its most recent state. Changes to the data made after the backup is created are not restored. Also, 4D temporarily stores record changes into a data cache before being flushed to the disk to optimize the performance of the application. However, when an incident occurs, this information is not yet saved permanently in the disk and can be lost. This is when using a log file comes into play.

Any changes done to the data such as modification, addition, and deletion including transactions are recorded in the log file. While record changes are saved in the data cache or saved in disk, the log file also records these changes chronologically. It will contain this information starting from the last backup performed on the database. Every time a backup is performed, the log file is backed up and the current log file is reset.

Using a log file, you can restore a previous backup and integrate changes that are made after the last backup was made. This brings the database up to date after an incident. Moreover if an incident occurs before changes in the data cache are flushed, this information are saved in the log file and any operations missing in the data file are automatically reintegrated the next time the database is opened (provided this option is selected in the Automatic Restore option of the backup preferences).

To create a log file, a backup should first be performed. You can then set *Use Log File* option in the Log Management section of the backup configuration setting as described in the Configuration section above. This option is selected by default when a database is created. You can stop the use of a log file at anytime by unchecking this option.

You can check the contents of the log file by selecting *Check Log file* item from the File menu. A log dialog is displayed as shown below.

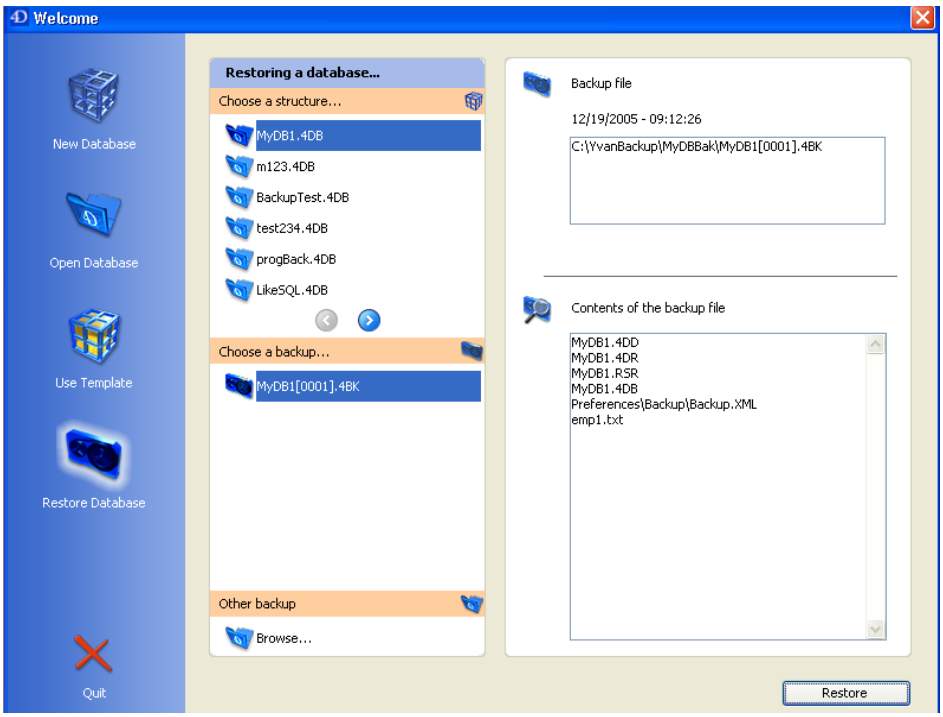


Restoring a Backup

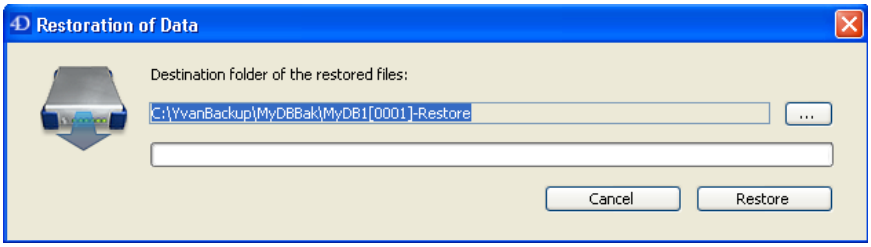
In case of incidents such as loss of data, crashes, or power outages, you can restore your backup manually or automatically. The log file can be integrated manually and automatically when restoring the backup as well.

Manual Restore

You can restore backup manually by using the Restore page of the Welcome dialog box. When you first launch the 4D Application (4th Dimension or 4D Server), you can click on the Restore Database button to bring up the restore page as shown below. You can also bring this up programmatically by using the RESTORE command in a method.



You can select a backup or backup log file to restore. The backup file path and its contents will be displayed on the right side of the restore page. To select a backup log file to restore, you will have to locate it using the Browse button in the Other backup section of the page. When you click on the Restore button, the Restoration of Data dialog, as shown below, is displayed. You can modify the destination folder of the restored files and start the restore by clicking on the Restore button.



Note: The backup file is named DatabaseName[000x].4BK (or any name you give your backup) and the backup log file is named DatabaseName[000(x-1)].4BL. For

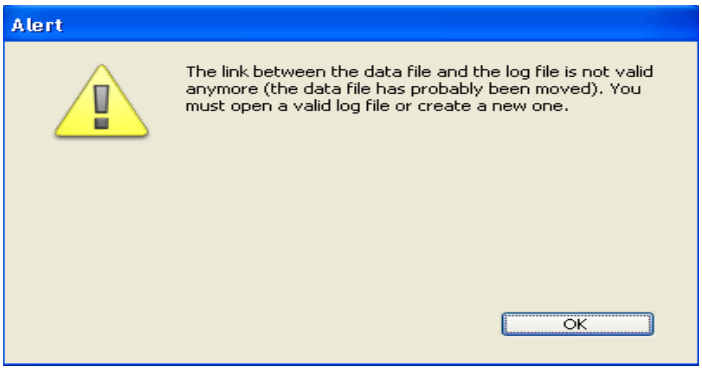
example if you have a database named DB1, the backup file will be named DB1[0001].4BK by default and the backup log file will be named DB1[0000].4BL. The backup log file number that is backed up is always one less than the backup file name number.

After restoring the database, a log file can be manually integrated to restore the changes made to the data after the last backup was made. The log file needs to correspond to the restored database in order for it to be linked. When a backup is made, the current log file is backed up and is cleared. Any changes made to the data after this will be recorded in the current log file. The backed up log file can then be used to link a backup made prior to the latest backup. Below are some scenarios to better illustrate this:

Scenario 1:

Backup is being performed everyday. On Wednesday, the database crashed and to restore a backup created last Tuesday and update it, the following steps are performed:

1) The backup created on Tuesday is restored.
2) When the restored database is opened, a dialog as shown below will be displayed.



3) When the OK button is pressed, it brings up an open dialog to select the log file to link. You can then select the current log file. This will integrate all the changes made to the data after the backup was made last Tuesday and before the database crashed.

Scenario 2:

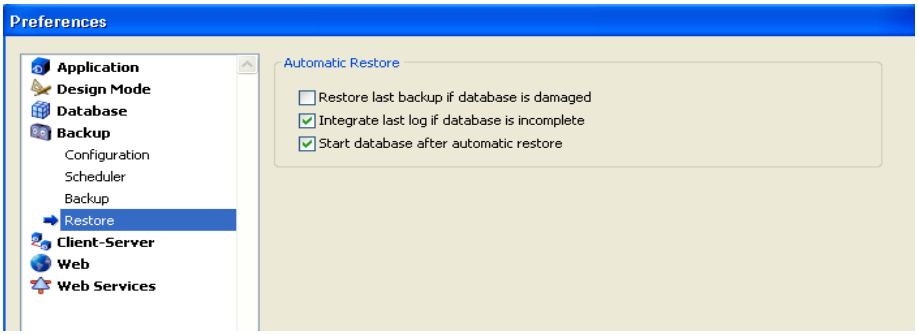
Yet still if you want to restore a backup created on an earlier date and make it up to date, a sequential restore should be performed. For instance just like in scenario 1 where a backup is performed everyday, to restore the backup made on Monday and integrate changes to the data before the database crashed on Wednesday, the following steps are performed.

1) Restore the backup made last Monday.
2) Restore the backup log file made last Tuesday.
3) Open the restored database and then select the restored log file. This should integrate changes to the data before the backup was made last Tuesday.

4) In the backup configuration page under the database preferences, uncheck the Use Log File option to stop the use of the log file.
5) Check the Use Log File option again and select the current log file that is used before the database crashed on Wednesday. This should integrate the changes made to the database after the database was made last Tuesday and before the database crashed. (If the database preferences cannot be accessed, you can move the log file that is used by the restored database to a different location. When you reopen it, you will be asked to select a log file. You can then choose the current log file to integrate the most current changes.)

Automatic Restore

You can automatically restore a backup when an incident occur by defining the restore procedures, as shown below, in the Restore page of the Backup theme.



The "Integrate last log if database is incomplete" and "Start database after automatic restore" options are selected by default. You can choose to restore the last backup if database is damaged.

If you have the "Restore last backup if database is damaged" option selected, a restore of the last backup is performed automatically when a problem with a database is detected when the current database is launched. And when the "Start database after automatic restore" is selected, the restored database is automatically re-launched. This does not require user intervention.

Accordingly with the option "Integrate last log if database is incomplete" selected, the current log file is automatically integrated as well when the restored database is open. Assuming you're currently using a log file and you have all the options selected above when an incident occurs, the last backup will be restored and when this restored database is re-launched, the current log file is automatically integrated to the database making it up to date. Backup operations performed are saved in the Backup Journal which is usually at the same level where the structure file is located.

Summary

4D 2004 has an integrated backup module. The backup and restore process can be done within 4D either automatically, manually, or programmatically. Each backup that is performed uses the backup settings and configuration that is set. The backup scheduler and the automatic restore options in the backup preferences allow you to perform backup and restore automatically. Backup can be performed manually through the File menu. The new backup commands allow you to programmatically start and manage a backup process. Security can be added to the data file by using a log file. Changes made to the data file including transactions are chronologically saved in the log file. This allows you to integrate changes to your database in case of incidents keeping it up to date. All in all, the features of the integrated backup module in 4D 2004 simplify the backup and restore operations of your database.