Mirroring with 4D Server v11 SQL

By Atanas Atanassov, Technical Services Team Member, 4D Inc.

Technical Note 09-16

Table of Contents

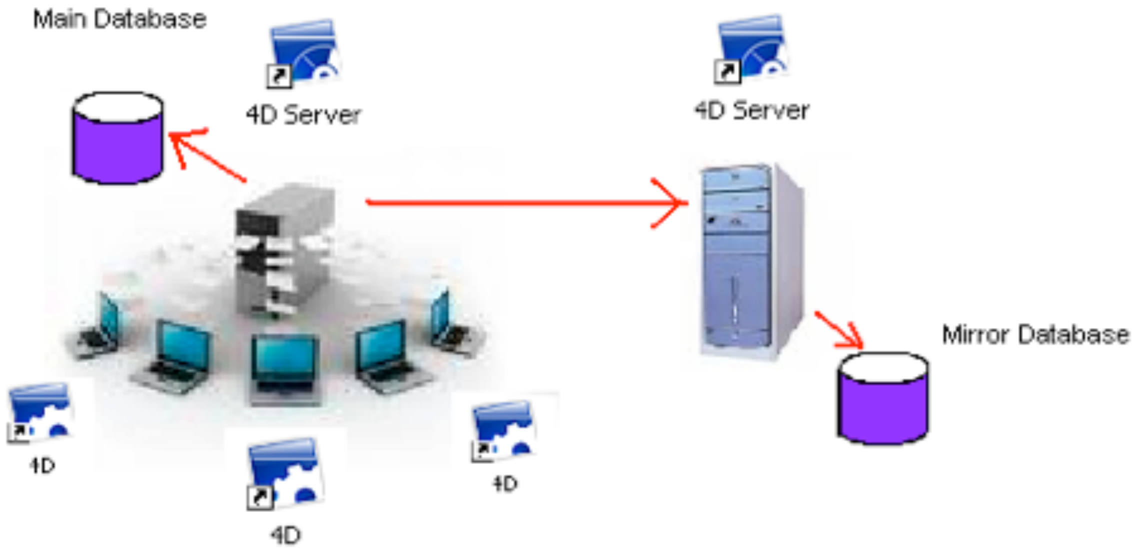
Abstract

Mirroring is a way to increase database availability. Database mirroring transfers 4D log files directly from one server to another. In case of failure, the client application can quickly connect to a mirroring server with no loss of committed data. This technology does not require proprietary hardware and is easy to set up and manage. This tech note includes a component that facilitates the addition of mirroring to any 4D database.

Introduction

In database mirroring a 4D v11 SQL instance sends a database's transactions to a copy of the database. The original database and server have the role of *principal*, and the receiving database and server have the role of *mirror*. The principal and mirror servers must be separate instances of 4D Server.

Here is a diagram to illustrate this:



The copy of the operational database is located on the mirror server and it is updated periodically. In this way, when there is an incident affecting the operational database, the mirror database can be used to get things back in working order quickly without any data loss. Moreover, the operational database is never blocked by a backup operation.

Without a logical mirror, the operational database is backed up on a regular basis. When a backup procedure starts in this situation, the database is in read-only mode. The good news is that with the new database engine in version 11, the time period that a database is in read-only mode due to a backup is very short, even for log files that are 2 gigabytes. Thus the operation can be programmed to take place outside of normal periods of database usage such as night hours or weekends.

However, there are certain organizations like hospitals, police and fire departments, where the database must be entirely operational 24 hours a day and during the weekend. Their databases can not be in read-only mode, even for a very short period of time. In these cases, setting a logical mirror is a solution.

Note: *Only data changes affect the mirroring database. Changes made to the database structure do not affect the mirror structure file. In other words, mirroring is not suitable for databases in the process of development and other structural changes.*

When a critical error occurs in the operational database, all transactions must be stopped during the recovery process. This may be a time consuming process involving troubleshooting, recovering techniques and potentially losing some of the transactions. With the mirroring model, the mirroring database can be launched as an operational database during the recovery process of the main database. The end users will not notice any changes or delays in their routine work.

Mirroring Model

The operational database is published on the principal 4D Server and an identical copy of the database is published on the mirroring 4D Server.

On the principal server, the log file is “segmented” at regular intervals. Since no backup procedure is carried out on the principal server, the operational database remains available at all times in read-write mode.

Each segment of the log file is sent to the mirror server, where it is integrated into the mirror database.

Note: *Any backup performed on the operational database will lead to the desynchronization of the operational and mirror databases thus breaking the mirror.*

A backup procedure can be started on the mirroring database. To avoid risk of desynchronization with the principal machine, 4D automatically locks the mirror machine when one of two operations occurs: Integration of the log file from the principal server or backup of the mirror database. Both processes are mutually exclusive.

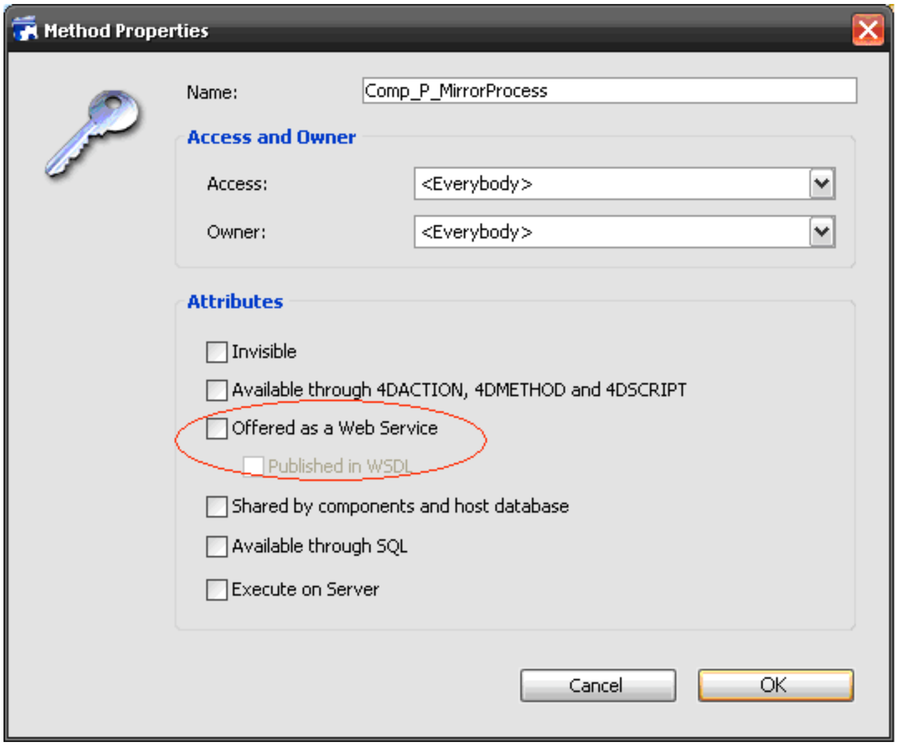
The mirroring server cannot modify data in the data file on its own. Also, nobody can login into the mirroring system and start a transaction. This is important because at this point, a backup procedure can be executed on the mirroring machine and will not desynchronize the mirror and principal databases.

For a better understanding of the mirroring process, the following sections explain 4D Web services and components. Only specifics in these topics that are related to mirroring are discussed.

More information can be found in the *4D Design Reference* manual and mirroring Tech Notes.

Soap Services – Review

In 4D, any project method can be published as a Web Service. In the method properties window, the check box "*Offered as a web service*" needs to be checked. Also, if "*Published in WSDL*" is checked, a WSDL file will be created in memory. This file carries information in XML format for the method name, URL, and SOAP parameters. Also, it publishes the web services in one of the available modes: RPC or DOC.



Name: Comp_P_MirrorProcess

Access and Owner

Access: <Everybody>

Owner: <Everybody>

Attributes

- □ Invisible
- □ Available through 4DACTION, 4DMETHOD and 4DSCRIPT
- □ Offered as a Web Service
  - □ Published in WSDL
- □ Shared by components and host database
- □ Available through SQL
- □ Execute on Server

Publication of Web services is carried out in three stages:

- Creation of the method to be published.
- Configuration of the publication.
- Publication itself.

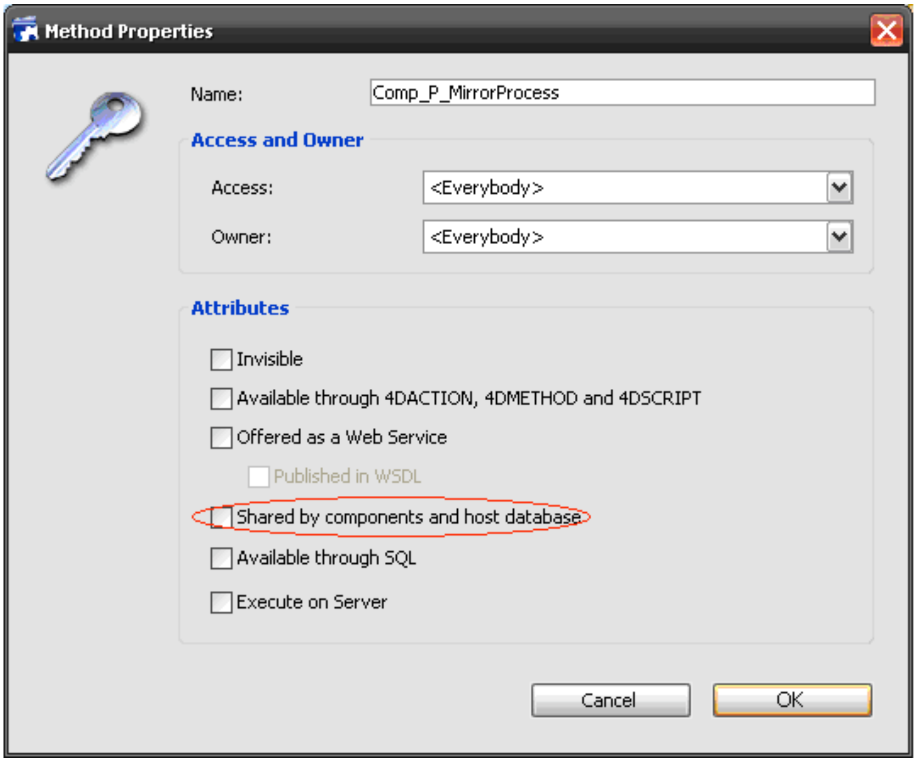
Note: *The publication of Web Services with 4D requires a specific license: 4D Web Services Server Expansion License.*

The 4D Web server automatically handles the management of the services and the WSDL file as well. For security reasons, in the *On Web Authentication* database method, there is a check for the method making a request to the mirroring server.

In this example, the name of the project method published as a WEB service is "*Mirror_SOAP_HandleEvents*".

Components in Version 11

Installation of components in version 11 is different than previous versions of 4D. The process of creating and installing a component is pretty straightforward. In the component, all methods that are to be accessible from the host database must be shared. This is done in the Method Properties window by checking "*Shared by components and host database"* option.



Method Properties

Name: Comp_P_MirrorProcess

Access and Owner

Access: <Everybody>

Owner: <Everybody>

Attributes

□ Invisible

□ Available through 4DACTION, 4DMETHOD and 4DSCRIPT

□ Offered as a Web Service

□ Published in WSDL

□ Shared by components and host database

□ Available through SQL

□ Execute on Server

When this option is checked, the name of the shared component method appears in the host database Explorer window and can be called from the host database. When the component is in interpreted mode (not compiled), the contents of the method is displayed in the preview area, but cannot be changed like the project methods in the Host database.

The component is placed inside the "*Components"* folder which is at the same level as the 4D structure file.

What is new for 4D v11 SQL

The following commands are necessary for building the mirroring model and have changed regarding the new component structure:

New Log File

This command does not accept any arguments. It closes the current log file (no further data changes are included), renames and creates a new current log file. The return value is the absolute path to the log file. New in version 11 is when the command is called from a component, the return value is the path to the log file of the Host database.

Note: *The "absolute path" to a folder is equivalent to the "full path" or the "long name" of this folder.*

Integrate Log File(pathName)

This command integrates the log file whose log name is passed as an argument into the current database. The current database in our example is the host database.

With the use of the "New Log File" command, the current log file is segmented on regular intervals. Then the log file segments have to be integrated sequentially in the same order as they have been created.

Both commands work only if they are executed on the server.

Log file name convention in 2004 and 11

The name convention in 4D 2004 is as follows:

myDatabase[0001 0003].4DL

Where the first four digits after the opening square bracket are the numbers of the backup file, and the second four digits are the segment numbers of the log file.

In version 11 the name of the log file is

myDatabase[0001 0003].journal

The numbers in the square brackets show that this is the third segment since the first backup procedure.

Requirements

For our example we use:

- Two machines running 4D Server. The number of servers depends on how many mirroring databases you are planning.
- At least one machine running 4D.
- A Web Service Server expansion license.

Setup the Logical Mirror

Installation of the component

Open the database where the mirroring component is to be installed or create a new database. Make sure that the “Use Log File” option in the Preferences, Backup theme is unchecked. Also, uncheck the “Data File” and “Structure File” in the “Backup Contents” pane for now and quit the database.

1. Place the component into the “Components” folder located next to the database structure file.
2. Place the “Macros V2” folder next to the structure file.

Installation of the Database methods in Host Database

Start the database and enter the following calls into the specified database methods:

1. On Startup – type “Mirror_install_OnStartup”.
2. On Web Authentication – type “Mirror_install_On_Web_Authentication”.
3. On Backup Startup - type “Mirror_install_On_Backup_Startup”.
4. On Backup Shutdown - type “Mirror_install_On_Backup_Shutdown”.
5. On Server Startup - type “Mirror_install_On_Server_Startup”.
6. On Server Shutdown – type “Mirror_install_On_Server_Shutdown”.
7. On Server Open Connection – type “Mirror_install_On_Server_Open_Connection”.

Entry points for the mirroring component

The dialog between the component and the host database is handled from the following three methods.

Method Name	Descriprion
Mirror_M_TestConnection	This method reads the mirror preferences stored into inter-process variables and verifies the connection between the main and mirror servers. It returns the connection status. • Connection established. • Connection failed. • Alert if the web services are running in demo mode and the connection time reaches the time limits.
Mirror_M_Preferences	This method launches the mirror preferences form. During the form’s “On Load” event, all mirror preferences are stored into inter-process variables.

Mirror_M_TransferLog	This is a wrapper method which creates a new process on the main server and sends a log file segment by executing *Mirror_LSendLogFile* in the context of the newly created process. This is needed only if the user wants to manually transfer log files between the main and mirror servers.

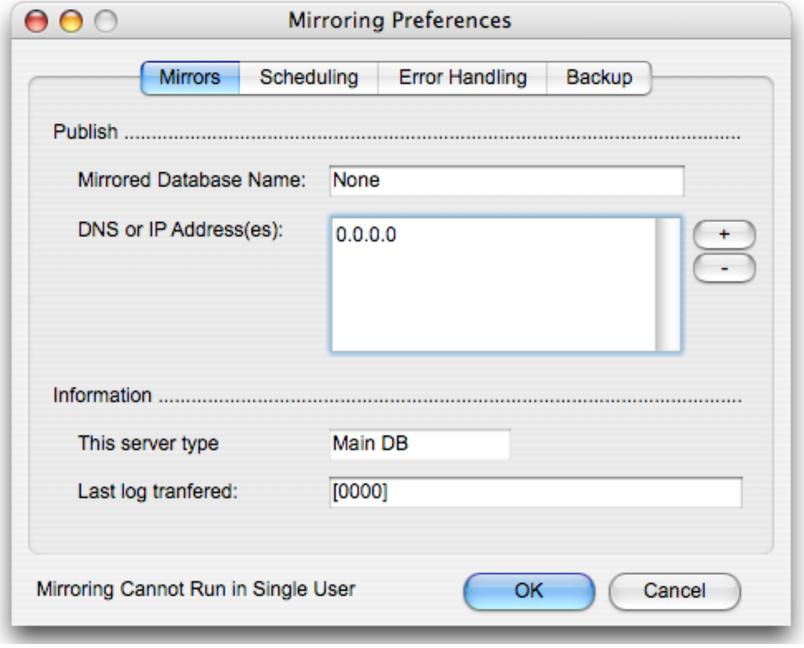
Setting up the preferences for the host database

With the first launch of the mirroring component, a file named “Settings.xml” is created in the database’s Resources/Mirror folder. The minimum requirements for users are entering the name of the mirror database and servers’ IP address(es). Mirror scheduling settings, error notifications and backups of the mirroring database are fully optional for the user although they are recommended for tracking the mirroring process and data protection.

The *Mirror_M_Preferences* method opens the “Mirror Preferences” window.

Mirrors Tab

Here the user specifies the name of the mirror database and IP address of the mirror server. These are the only necessary settings to set up mirroring between two 4D servers.



Publish

Mirrored Database Name: None

DNS or IP Address(es): 0.0.0.0

Information

This server type: Main DB

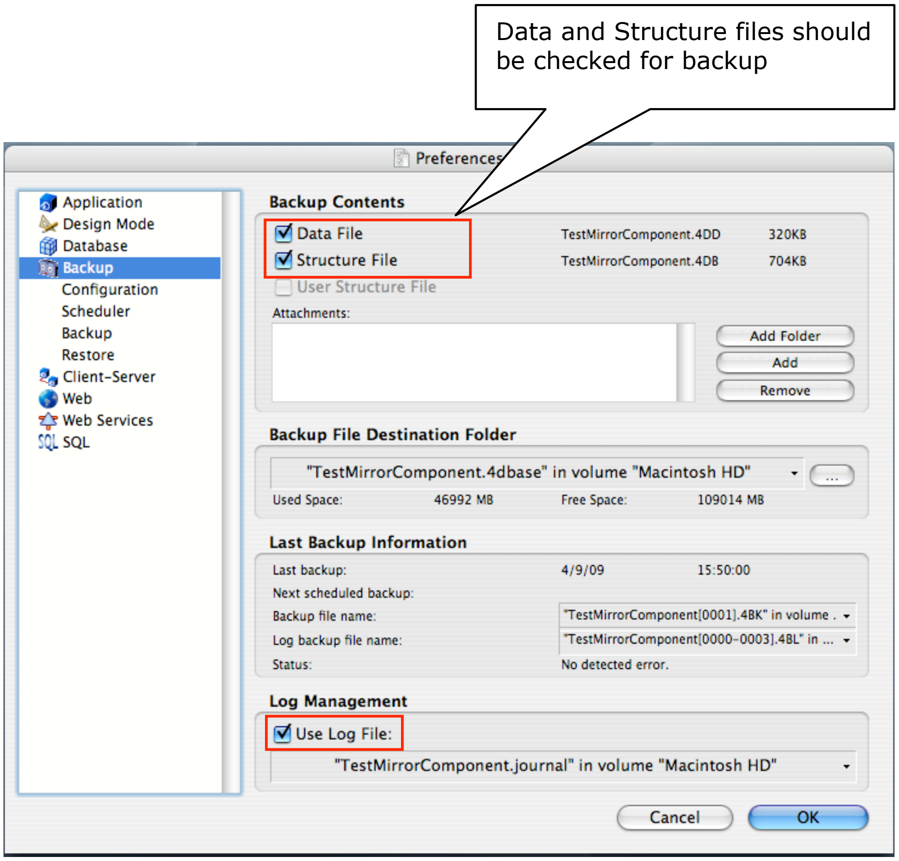
Last log tranfered: [0000]

Mirroring Cannot Run in Single User

1. In the “Mirrored Database Name” field, type the name of your database and in the “DNS or IP Address(es)” field, type the IP address of the mirroring server(s) followed by column and the port number where the

web services are published. The port number can be obtained from the database preferences in the Web theme.

2. Backup the database with the Log File option checked. This is done from the Backup theme in database Preferences.



Data and Structure files should be checked for backup

Backup Contents

☑ Data File TestMirrorComponent.4DD 320KB
☑ Structure File TestMirrorComponent.4DB 704KB
□ User Structure File

Attachments:

Backup File Destination Folder

"TestMirrorComponent.4dbase" in volume "Macintosh HD"

Used Space: 46992 MB Free Space: 109014 MB

Last Backup Information

Last backup: 4/9/09 15:50:00
Next scheduled backup:
Backup file name: "TestMirrorComponent[0001].4BK" in volume .
Log backup file name: "TestMirrorComponent[0000-0003].4BL" in ...
Status: No detected error.

Log Management

☑ Use Log File:
"TestMirrorComponent.journal" in volume "Macintosh HD"

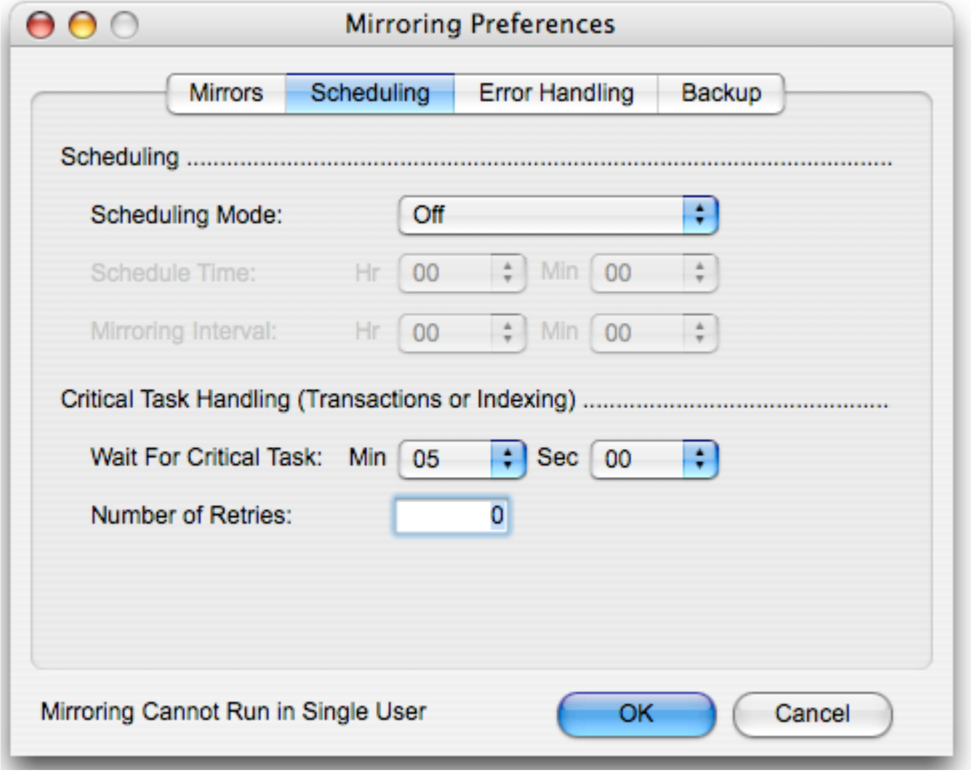
3. The log file should be saved into the database folder. An alert message is displayed from step 2: "This database is part of a mirror. Continuing the backup will break the Mirror!" Go ahead and press "Break the Mirror" button.
4. Copy the database folder and move it to the mirroring machine.
5. Find the "Setting.xml" file located in the "MyDatabaseFolder/Resources/Mirror" folder. This file contains the mirroring preferences for the main server. Since a copy of the database has been moved to the mirroring machine and the mirroring server has its own preferences, the user should delete the Mirror folder and its contents.
6. Open the database with 4D Server. Another confirmation window shows up and asks for the type of the server. Choose "Mirror".
7. Check to see if the Web server is running. If not, press "Start the server". You can also check the option "Publish Database at Startup". This option is located inside Web theme in Preferences.
8. Open the database with the principal (Main) server.

Use 4D as a client to connect to the principal server. At this point you have set up the mirroring system.

The Project form presented earlier also handles the tasks for scheduling, error messages and backup.

Scheduling Tab

The time and interval settings for mirroring are in the Scheduling tab. It has four options to choosing from. The lower half of the page is to set the time and the number of retries for creating a new log file during critical operations like transactions and indexing.


Mirroring Preferences

Scheduling

Scheduling Mode: Off

Schedule Time: Hr 00 Min 00

Mirroring Interval: Hr 00 Min 00

Critical Task Handling (Transactions or Indexing)

Wait For Critical Task: Min 05 Sec 00

Number of Retries: 0

Mirroring Cannot Run in Single User

- "Sheduling Mode" options are:
  - Off – The scheduling mode is disabled.
  - Time Only - Set a specific time to send the log file once in 24 hours
  - Time & Interval - Sends the log file at the specific time and then repeats at the selected interval.
  - Interval Only - Repeats mirroring (sends the log file) in specific intervals in hours or minutes.
- "Wait For Critical Task" specifies the time period for trying to create a new log file. Retry intervals are specified in the "Retry after" option in the database Preferences, Backup theme.

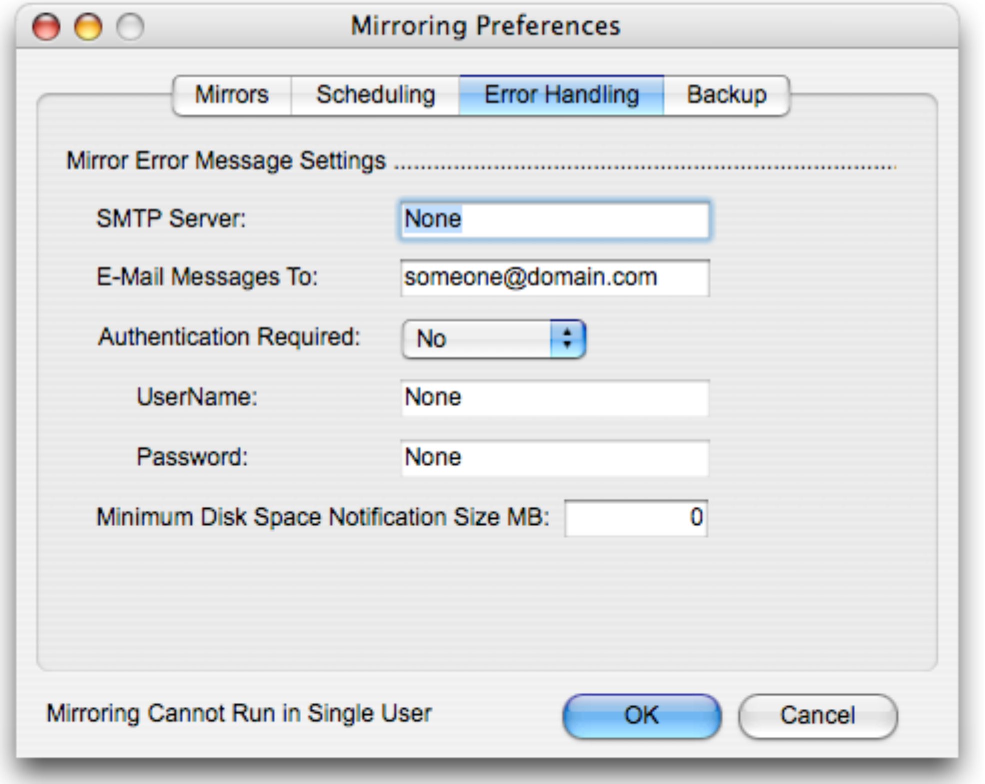
Note: *The time in "Retry after" option should be less than "Wait For Critical Task" time. "Retry after" recommended value is one second.*

- "Number of Retries" sets how many times the server tries to create a new log file. If it fails to create the log file during the scheduling time, mirroring will be skipped till the next time.

Error Handling tab

All errors and alerts regarding the mirroring procedure are handled from the component. It is very important that the user receive feedback from the main or mirror server if something goes wrong. Examples include creation of the new log file fails, connection problems or failure to integrate the log file. Without such notifications the user will not be aware if the mirroring accidentally fails and stops.

In this page are the settings for error notification system built with the mirroring component.



Mirror Error Message Settings

SMTP Server: None

E-Mail Messages To: someone@domain.com

Authentication Required: No

UserName: None

Password: None

Minimum Disk Space Notification Size MB: 0

Mirroring Cannot Run in Single User

- "SMPT Server:" - Specify the IP or the name of the SMTP server. If this setting is left to "None", no messages will be sent.
- "E-Mail Messaged To:" Specify the email where all error notification should be sent. This option utilizes the 4D Internet commands plug-in.

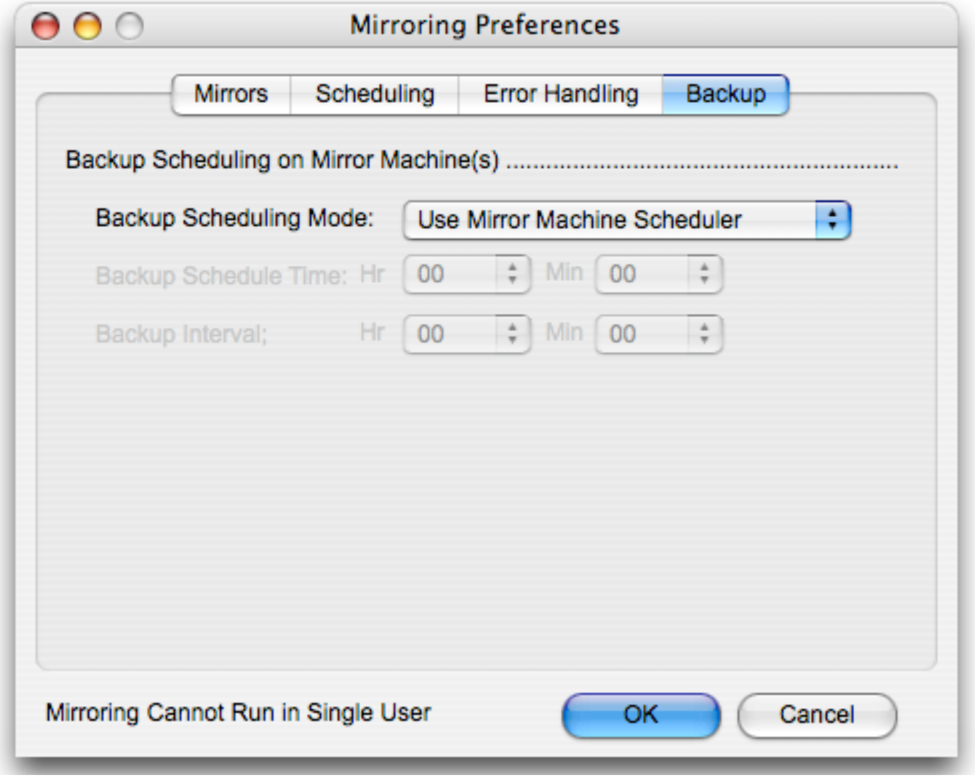
Note: *The Plug-ins for the component in 4D v11 SQL must be installed in the "Plug-ins" folder of the Host database.*

- "Authentication Required:" - UserName and Password fields are standard settings for the email box. They can be obtained from the account settings information.
- "Minimum Disk Space Notification Size:" Specify the size of the free space dedicated to the main server. Over time, the number of log files will increase and this increases the amount of used disk space. When it reaches the size specified in the "Minimum Disk Space Notification Size MB" option, it triggers an email with an alert message for reaching the disk limit. This option does not check the mirror machine, because backup routines can be performed only on the mirror server and the backup code takes care of deleting these log files.

Backup Tab

It is very important to remember that backup procedures are started only on the mirroring server. A backup on the main server would desynchronize the databases and "break the mirror".

In the "Backup" tab are settings for backups on the mirror server.



Backup Scheduling on Mirror Machine(s) ....

Backup Scheduling Mode: Use Mirror Machine Scheduler

Backup Schedule Time: Hr 00 Min 00

Backup Interval; Hr 00 Min 00

Mirroring Cannot Run in Single User

Backup Scheduling Mode has four options to schedule the backup routine on the mirroring server.

- "Use Mirror Machine Scheduler" – The backups occur using the Backup preferences settings on the mirrored server.
- "Via Main Server Time" – This option sets up the time for backup from the main server.

The last two options work in the same way as described in the previous section.

- via Main Server Time & Interval
- via Main Server Interval

Flow of the Mirroring mechanism

During the server startup, the mirroring component reads the preferences and stores these preferences into interprocess variables. These preferences control the mirroring process on the main server. They were saved in the "Setting.xml" file located in the Resources/Mirror folder during the initial setup.

When both servers are up and running, the user needs to verifiy the connection between them. This is very important because there is no need to create a log file or try to move it if the connection fails to begin with. Testing the connection is done by executing the *Mirror_M_TestConnection* method. Security connection between both servers is handled from the *Mirror_OnWebAuthentication* method which is installed in the On Web Authentication database method.

```
`Mirror_OnWebAuthentication


C_Boolean($0)

Case of
   :(Not(Is SOAP request))
         $0:=False
   :(Get SOAP info(SOAP method name)#"Mirror_SOAP_HandleEvents")
         $0:=False
   :Else
         $0:=True
End case
```

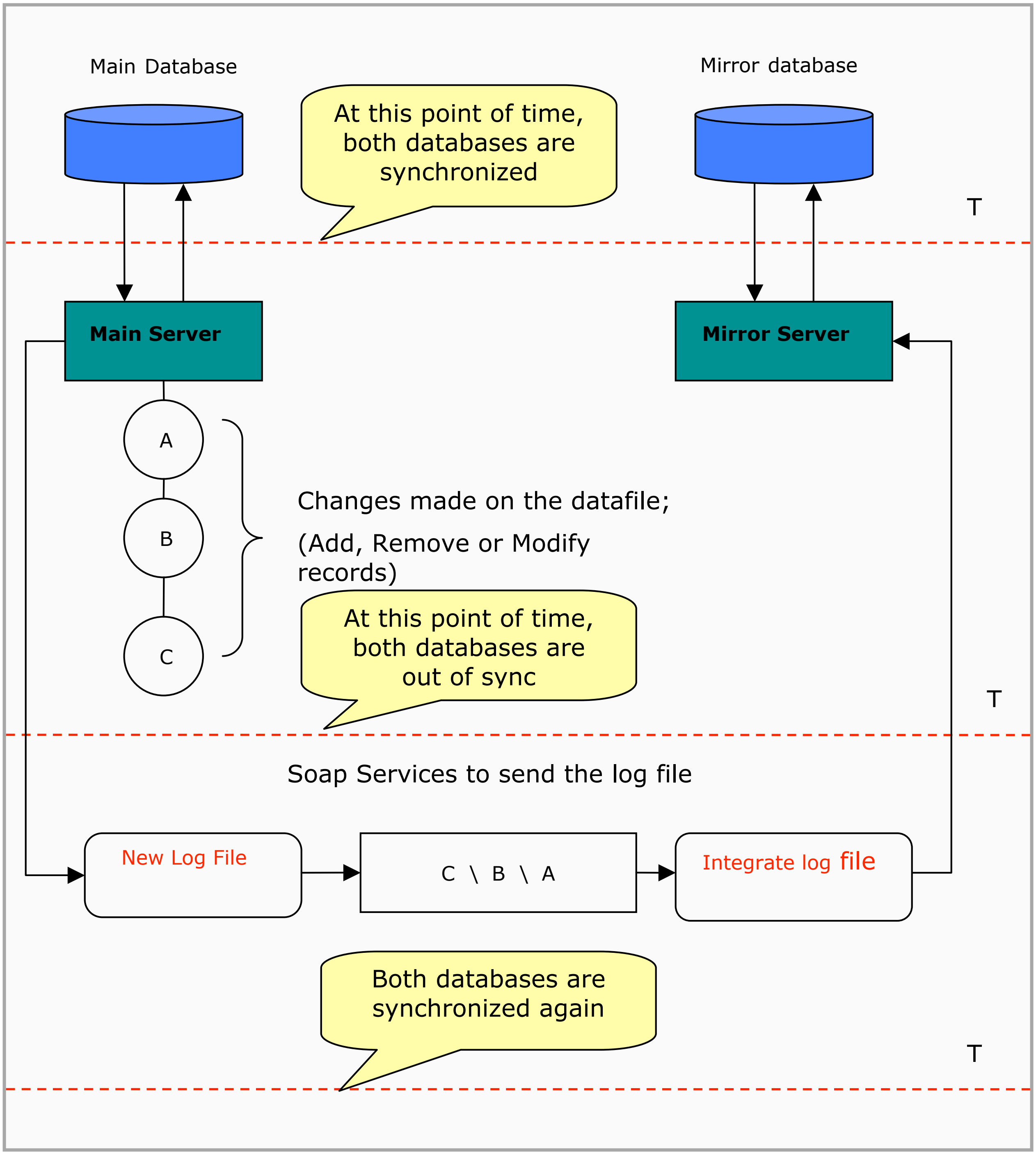
This method returns False if the web request is not a Soap request. If it is a Soap request, it guarantees that this request is done from the *Mirror_SOAP_HandleEvents"* method. All other Soap request are rejected.

There are actually three Soap calls from the principal to the mirror server.

The first call verifies if the mirror database is running and if it is the actual mirror database. The actual transfer is done during the second Soap request. The content of the log file is put into a BLOB and transferred to the mirror database.

The third Soap call verifies the integration status of the log file and returns this status to the main server. At this point the mirror database has been successfully updated.

This is a flow chart of the mirroring process.


Main Database
Mirror database
At this point of time, both databases are synchronized
T
Main Server
Mirror Server
A
B
C
Changes made on the datafile;
(Add, Remove or Modify records)
At this point of time, both databases are out of sync
T
Soap Services to send the log file
New Log File
C \ B \ A
Integrate log file
Both databases are synchronized again
T

Note: *For more detailed information about the design please, refer to the previous two Tech Notes for Mirroring.-"Mirroring with 4D 2004.4, PartI and PartII", knowledgebase assets 44173 and 44242.*

Conclusion

Creating a mirroring system is not a complicated task. The heart of the mirroring mechanism is the log file and both "New Log File" and "Integrate Log File" commands. Once the log file has been created it can be integrated into the copy of the main database. This copy of the main database can be used as a backup or for executing time consuming queries. In other words the mirroring system can be used not only for protecting data but also for increasing the performance of the main database.

It is not a necessity to have a mirroring system, but it is one very convenient way to bring your system up in case of an accident with your main database.

Related Resources

- Technical Note: Mirroring with 4D 2004.4, Part I
- http://kb.4d.com/search/assetid=44173
- Technical Note: Mirroring with 4D 2004.4, Part II
- http://kb.4d.com/search/assetid=44242
- 4D v11 SQL Language Reference
- http://www.4d.com/support/documentation.html