

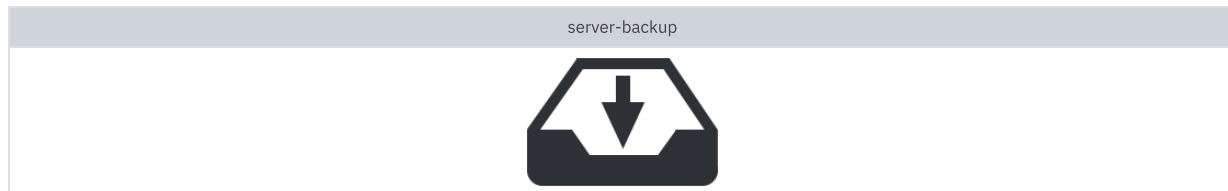
FileWave Server Backup and Restore

Summary

Backing up your FileWave server is always a good idea. Back in '97, a friend of mine spilled a pot of coffee on our server rack; you never know when bad things can happen to your hardware. Now, it's bad enough to lose the hardware, but think of all the hours represented by those perfected, irreplaceable FileWave Filesets! In order to prevent this disastrous loss, we have created a script to automatically back up your FileWave Server's database and data files, you can also run these scripts manually if you want.

Current Version

Download macOS / Linux - Version 4.5.4 (Jan 24, 2024)



i If you ever want to see the changes that have been made to this script over time please check out the GitHub repo - [ilevitsk/filewave-serverbackup: The script that can backup and restore a FileWave Server \(github.com\)](https://github.com/ilevitsk/filewave-serverbackup)

i Due to the nature of the FileWave databases, using active-backup solutions, such as Time Machine or CrashPlan, can corrupt the FW DB.

w Alternate File Systems - Windows and Unix systems treat permissions in completely different ways. If attempting to backup from, for example, a Linux server to a Windows share, data will be backed up successfully, however, permissions will not be preserved. Permissions can and will need to be reset on files after a restore is actioned, and the script to repair permissions is located in the [FileWave Server Repair Permissions KB article](#).

Updating the backup script

If you already have a backup scheduled on macOS or Linux, replace the script you are already using with the new version shown.

First, let's find out where your existing backup script is located:

```
sudo crontab -l|awk {'print $6'}
```

The above shows you a line like :

```
/root/scripts/backup_server_osx_linux.sh
```

You'll probably want to check if the script is newer than what you are using. Use the below command and replace `/path/to/` with the path to your script you found in the prior step. You can see the version number of the latest script at the top of this article. Be aware that we only added the script version to the script in 2024 so if no version comes back or some weird text then you can be certain you need to update.

```
awk '/version=/' /path/to/backup_server_osx_linux.sh
```

Now that you are sure you want to update it, the below will show you where to place the file on your server. Please replace `'/root/scripts/backup_server_osx_linux.sh'` in the last two lines of the instructions below with what was returned to you by the last command before running the following:

```
wget https://kb.filewave.com/attachments/188 -O backup_server_osx_linux.sh.zip
unzip ./backup_server_osx_linux.sh.zip
# Remember the next 2 lines need to reflect where the script is going...
mv ./backup_server_osx_linux.sh /root/scripts/backup_server_osx_linux.sh
chmod a+x /root/scripts/backup_server_osx_linux.sh
```

w Note: If you move the Data Folder to another network share, please make sure to update the path as well in the Backup Script.

On Mac OS X and Linux

Warning for macOS - With the added security of macOS Catalina (10.15) and beyond, you will need to give Full Disk Access to Cron, the utility used to run the automated backups. To do this drag and drop Cron (/usr/sbin/cron) into System Preferences > Security & Privacy > Privacy > Full Disk Access

Before you start, please ensure that the script has the execution permission (chmod 755).

The script is used to schedule backups or to run a backup manually. The first parameter passed to the script (required parameter) is "setup" or "run"

Automatic Backup

Use the "setup" parameter to schedule a backup task to run either weekly or daily. The second parameter options are [weekly | daily] and the third is the path to the back folder. For Big Sur servers, be sure the Terminal.app and /usr/sbin/cron have Full Disk Access to allow the cron job to complete successfully.

Examples: (Please use the full path to the script)

```
/path_to_script/backup_server.sh setup daily "/Volumes/backupFolder"  
/path_to_script/backup_server.sh setup weekly "/Volumes/backupFolder"
```

Manual Backup

Use the "run" parameter to run the backup manually right now. The second parameter is the backup folder, and the third parameter is the task mode "manual". The script has to know that it is running manually and not from a cron job.

Examples:

```
/path_to_script/backup_server.sh run "/Volumes/backupFolder" manual
```

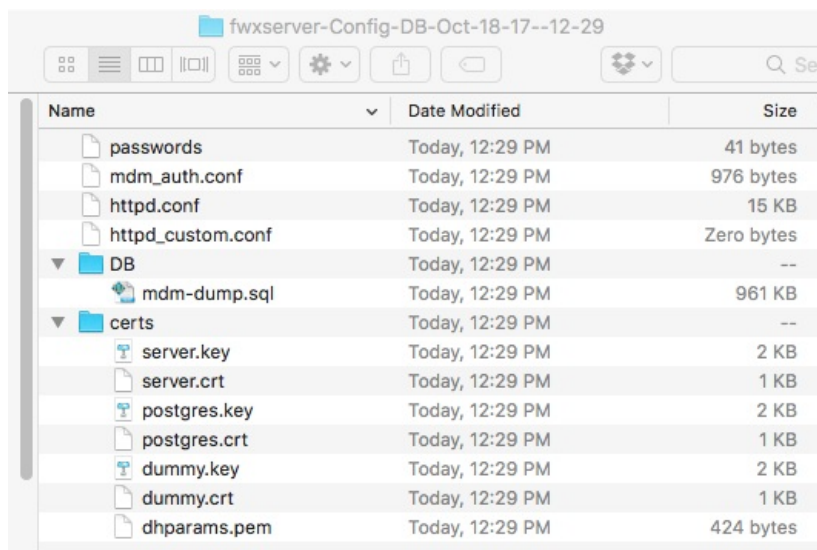
Backup log file

The log is saved by default to "/private/var/log/fw_backup.log"

If you want to change the path, edit the script and change the variable "logFile".

Checking Automatic Backups

Please make sure to check your Automated backups periodically to make sure they are still running successfully. The main thing to check is to unzip your most recent zip folder created by the backup and check that inside, you see a "certs" folder, "DB" folder, httpd.conf, httpd_custom.conf, mdm_auth.conf, and a passwords file. You would also want to make sure that inside the DB folder that is created, you see a "mdm-dump.sql" file, and the size of the file is not showing as 0 bytes.



Name	Date Modified	Size
passwords	Today, 12:29 PM	41 bytes
mdm_auth.conf	Today, 12:29 PM	976 bytes
httpd.conf	Today, 12:29 PM	15 KB
httpd_custom.conf	Today, 12:29 PM	Zero bytes
DB	Today, 12:29 PM	--
mdm-dump.sql	Today, 12:29 PM	961 KB
certs	Today, 12:29 PM	--
server.key	Today, 12:29 PM	2 KB
server.crt	Today, 12:29 PM	1 KB
postgres.key	Today, 12:29 PM	2 KB
postgres.crt	Today, 12:29 PM	1 KB
dummy.key	Today, 12:29 PM	2 KB
dummy.crt	Today, 12:29 PM	1 KB
dhparams.pem	Today, 12:29 PM	424 bytes

Restoring from Backup

The day may come that you need to restore from backup. This is something you should probably include [Technical Support](#) on


because if you are at that point then something bad has probably happened. Knowing where to put the files from the backup is easiest found in our [Migrating your On-Premise FileWave Server to new Hardware](#) article because it tells you where to move the backup files when restoring and how to bring the database back.

Legacy Backup Script Versions

FileWave Server Backup Script for version 3.x	**FileWave Server Backup Script for version 4.x**	**FileWave Server Backup Script for version 5.6.x or older**
999.backupFWXServer.sh.zip	999.backupFWXServer_v4.sh.zip	999.backupFWXServer_v5.sh

FileWave Server Backup Script for versions 5.7 through 10.1.1 (compatible with Mac and Linux)	FileWave Server Backup Script for versions 5.7 through 10.1.1 (Windows)	FileWave Server Backup Script for 10.1.1 through 13.x (Windows)
backup_server_osx_linux.sh.zip	backupScript.zip	backupScriptv2.zip

On Windows

 Windows Server is no longer supported but the below is included for [migration](#) purposes. Use a legacy version of the backup script available above.

The Windows script takes two parameters, the path to the backup folder and then the frequency, which can one of these three options [daily | weekly | now]

Automatic Backup

Here is an example that shows how you can schedule a daily backup task. (Please use the full path to the script)

```
C:\_path_to_backup_\backup.bat "D:\MyFolder\FileWave Backups" daily
```

Manual Backup

You can run the backup manually by providing the "now" parameter. This can be useful before you upgrade your FileWave server to a newer version.

Example: (Please use the full path to the script)

```
C:\_path_to_backup_\backup.bat "D:\MyFolder\FileWave Backups" now
```