

Backups! More Backups!

Maximize the number of available backup sets ...

By Wim Decorte

A big Thank You goes to Rob Russell (SumWare Consulting Limited, www.sumware.co.nz) for providing the Mac OSX shell script. Rob is a true FileMaker guru and I'm very fortunate to be included in his circle of friends.

Preface

Version 1 of this article was written when FileMaker Server 7 for Windows could actually run VBScripts. At some point in the update cycle of the Windows version of FileMaker Server 7 and on into FileMaker Server 8 that functionality unfortunately went away.

At this point in time (with revision 4 of FileMaker Server 7 and revision 1 of FileMaker Server 8) you can still select VBScripts in the FileMaker Server Scheduler, but they will not run. At runtime FileMaker Server will log an error saying that the script could not be found and will disable the schedule.

FileMaker Server for Windows can now only execute Windows batch files (files with the extensions BAT and CMD). You can easily wrap a VBscript into a batch file though as we will show in the article.

Keep in mind however that there is one important gotcha about the FileMaker Server scheduler: everything it executes runs in the security bubble of the account that runs the FileMaker Server services. The FileMaker Server Scheduler therefore will fail to execute anything that requires privileges that the FileMaker Server services does not have. Copying or moving backup sets across the network for instance is impossible with the default "Local System" account used by FileMaker Server.

I *do not* recommend changing that default service though just to make FileMaker Server schedules work. That could potentially lead to all sorts of security issues. I do recommend using the Windows Task Scheduler for anything that goes beyond the local FileMaker Server machine.

Enjoy this article.

You can never have enough backup sets. The more backups you have close at hand the quicker you can get back up and running and minimize the unproductive downtime of your users, or having to redo all the development work you just lost.

Easy backups without interrupting the user sessions is one of the main features of FileMaker Server. Let's explore how we can use that feature to maximize the number of backup sets readily available with the minimum amount of setup.

So here's the challenge: take hourly backups and have each hourly backup set available for any day of the past. Ideally a date/time stamped backup set so you can quickly find the one that you want to restore. How would you go about doing this?

One approach – and probably the one that comes to mind first - would be to configure one FileMaker Server backup schedule for every hour of the day. For this we would need to create one folder for every backup set. A folder named 0600 for the 6am backup, a 0700 folder for the 7am backup and so on. Next we would need to set up a bunch of schedules that back up all files to their proper folder. A schedule named 0600 to back up all the hosted files to the 0600 folder (see Figure 1), a 0700 schedule to put the files in the 0700 directory and so on.

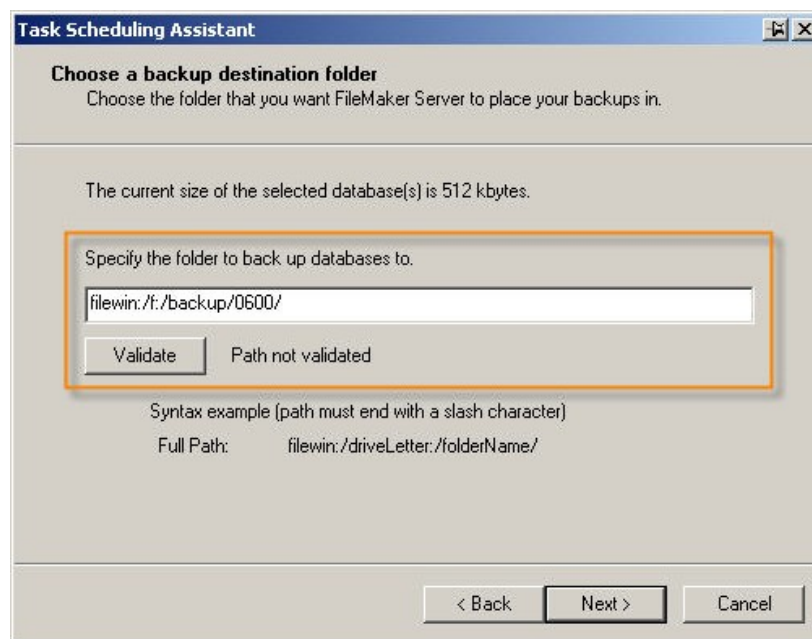


Figure 1 - Setting the path to where the backup is saved. That folder needs to exist already.

The end result would be a set of schedules like in Figure 2:

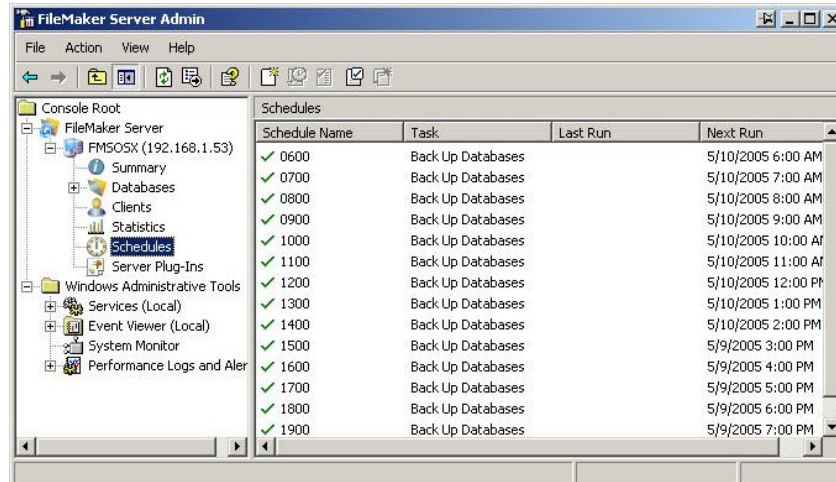


Figure 2 - All schedules necessary to make hourly backups that do not overwrite the previous backup.

But this is only half of what we've set out to do: now we have to make sure we have backups of all these folders for every day of the past. Those backups would be made with regular backup software to some other medium (Tape, CD, DVD, zip drive, external hard disk,...). The external backup software would date/time stamp the backup sets.

You'll agree with me that it takes quite a bit of work to set this up

- create the folders for the backups
- create all the individual schedules
- set up the additional backup to backup the backup folders

When it comes to restoring quickly, you'll only have the backups of the last 24h available. The rest will have to come from whatever backup medium made the second backup to.

Let's look at another approach that requires a lot less work and produces those hourly backups in a way that they can be easily restored.

Note that the approach outlined here is for Windows. But the same logic applies to OSX where shell scripts can be used instead of VBscripts. At the end of the document you will find a Mac OSX shell script that does the same thing.

We start off by creating one normal backup schedule to an existing folder (see Figure 3)

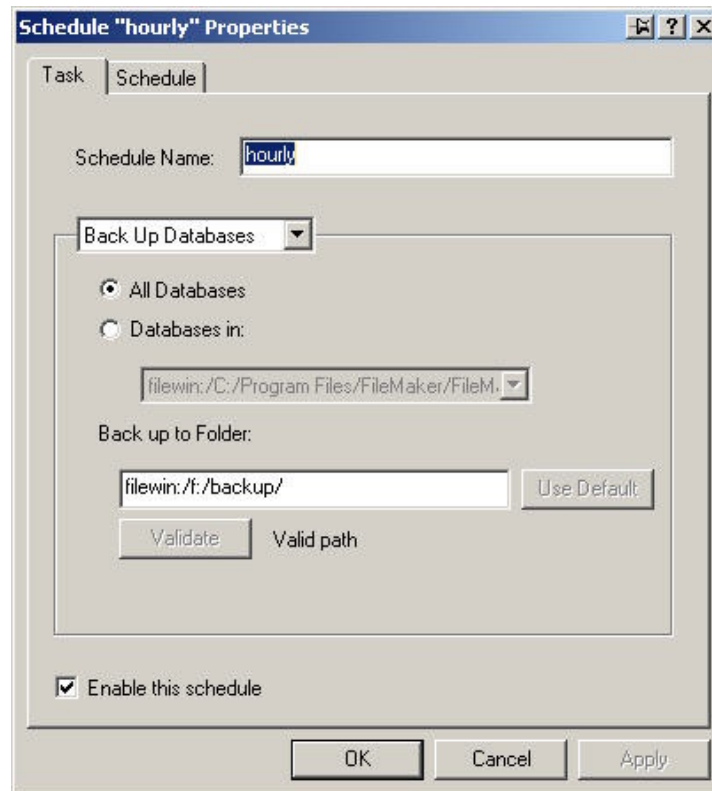


Figure 3 - A normal backup schedule to an existing folder.

And in the "schedule" properties we set it to repeat ever hour (Figure 4).

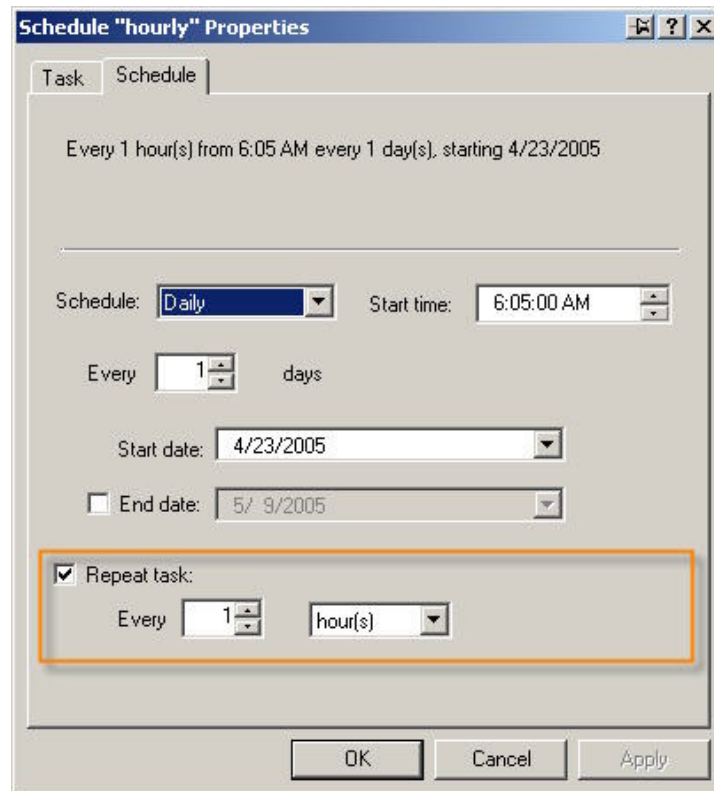


Figure 4 - Make the schedule repeat every hour.

Each time this schedule runs it will effectively overwrite the previous backup, so how does that get us where we want to be: saving a date/time stamped backup set every hour of every day? We need a second schedule to do this. But instead of backing up files, this schedule will run a small VBscript to create the date/time stamped archive file (Figure 5) and we schedule that one to run every hour too, some time after the regular backup schedule (20 minutes later in Figure 6). You'll remember from the preface though that the FileMaker Server Scheduler can not run a VBscript directly. We need a batch file to run the VBscript for us.

A one-line batch file named "FMSbackupZip.CMD" will do:

```
cscript.exe "C:\Program Files\FileMaker\FileMaker Server7\Data\Scripts\FMSbackupZip.VBS"
```

Note that we're using quotes around the path because the path contains spaces. "cscript" is the command line based script host that can run VBscript. You don't really need to call cscript¹; normally Windows will be smart enough to figure out what application to run by looking at the VBscript extension. But having said that; you get better results by explicitly saying what you need done instead of relying on what is understood...

¹ - "cscript" is the command line version of the Windows script host; a Windows system executable that runs VBscripts (and some other scripting languages like Jscript).

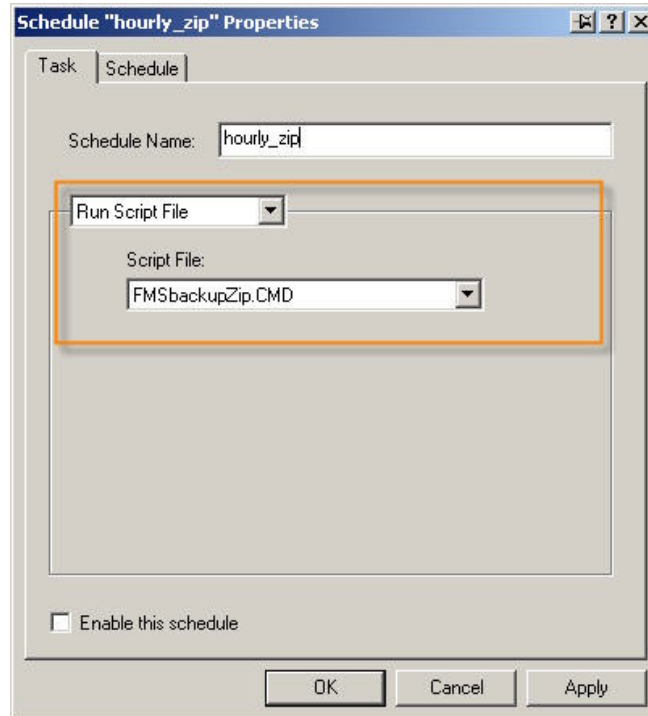


Figure 5 - Schedule a batch file that calls a VBscript to create the date/time stamped zip file.

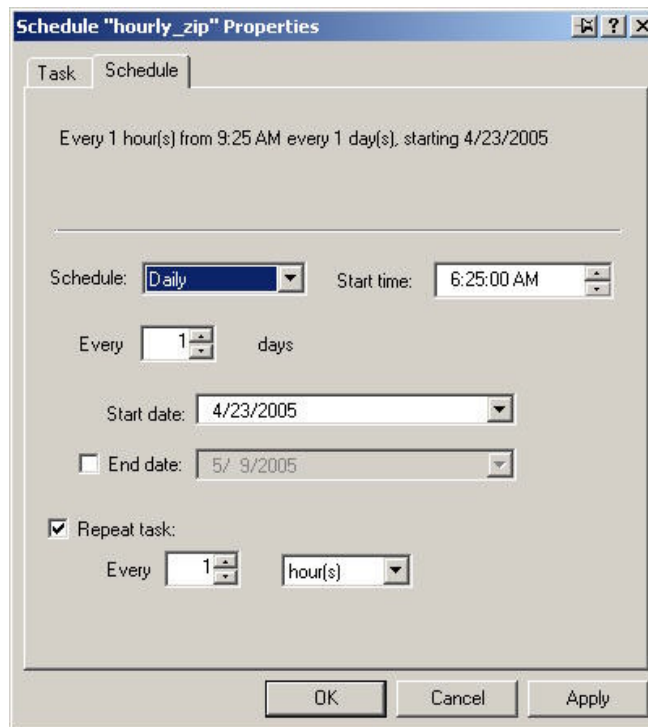


Figure 6 - Make the schedule that runs the batch file to repeat every hour.

The VBScript that gets called by the batch file is listed below and is very simple. What it does is run the WinZip command line utility and instruct it to create a zip archive of the FileMaker files in the folder that FileMaker Server backs up to. The VBScript is heavily commented (in green) so you can follow what it is doing.

```

1.  q = Chr(34) ' the quote character
2.  strZip = "cmd /c " & q & "C:\Program Files\WinZip\wzip.exe" & q & " -
    m -ee "
3.  ' command line string to invoke WinZip, -m to delete the originals, -
    ee For maximum deflation. See WinZip help file for all options.
4.  Const SourcePath = "F:\backup\"
5.  ' this is the path where FMS makes its backups to!!
6.  Const DestinationPath = "F:\backup\FMS_backups\"
7.  ' Backup folder to put the zip file in
8.  ' build the date/time stamp
9.  ' this format: 20050509_0600
10. TheDate = Year(Now()) & _
        Right("00" & Month(Now()), 2) & _
        Right("00" & Day(Now()), 2) & "_" & _
        Right("00" & Hour(Now()), 2) & _
        Right("00" & Minute(Now()), 2)
11. ' build the complete command line to execute
12. runZip = strZip & DestinationPath & "FMS_" & TheDate & ".zip " &
    SourcePath & "*.fp7"
13. Set SH = WScript.CreateObject ("WScript.shell")
14. SH.run runZip
15. ' prepare to copy the file (write filename to registry, another script
    will later copy the file
16. sh.RegWrite "HKLM\SOFTWARE\FMS_backup\", DestinationPath & "FMS_" &
    TheDate & ".zip", "REG_SZ"
17. Set SH = Nothing
18. WScript.Quit

```

Lines 2 to 12 basically create a string like this²:

```
cmd /c "C:\Program Files\WinZip\wzip.exe" -m -ee F:\backup\FileMaker
Server_backups\FileMaker Server_20050509_0600.zip F:\backup\*.fp7
```

² - We need the quote character (line 1) in line 2 because there is a space in the path to WinZip. Because of that space we need to enclose the path in quotes to have it treated as one string.

which translated, means:

create a zip file named "FileMaker Server_20050509_0600.zip" (current date & time of course) in folder F:\backup\FileMaker Server_backups\, and put in it all the FP7 files from the F:\backup\ folder. Delete the original files when done ("-m") and use maximum deflation to save space ("-ee").

These last two are WinZip options and are very well documented in the WinZip help file. The "cmd /c" at the start of the string launches the Windows Command Interpreter (the black "DOS" box) and tells it to quit when it is done executing the command.

Lines 13 and 14 tell the Windows' Shell to run that command line.

Note that I'm using the WinZip command line utility to do this (<http://www.winzip.com/prodpagecl.htm>). But you can just as easily use WinRar (http://www.rarlab.com/rar_archiver.htm) or the Open Source 7-zip (<http://www.7-zip.org/>) or any other archiving utility as long as it has a command line or an ActiveX interface. It needs either a command line syntax or an ActiveX interface because it should not rely on a GUI: there is no GUI if nobody is logged into the machine.

These two schedules produce an unlimited set of date/time stamped backup sets, with no elaborate setup: no need to create separate folders for each backup set, no need to set up a separate schedule for each hourly backup.

Coming back to the security perspective I mentioned in the preface, if the ZIP schedule fails then it probably means that the FileMaker Server service through the account it is running under does not have the necessary privileges for the folders involved. In that case I suggest leaving the privileges as they are and to use the Windows Task Scheduler³ instead of the FileMaker Server Scheduler. In the Windows Task scheduler you can specify a set of credentials with sufficient privileges that will be used for the task.

³ - To get to the Windows Task Scheduler: Start ► Control Panel ► Scheduled Tasks

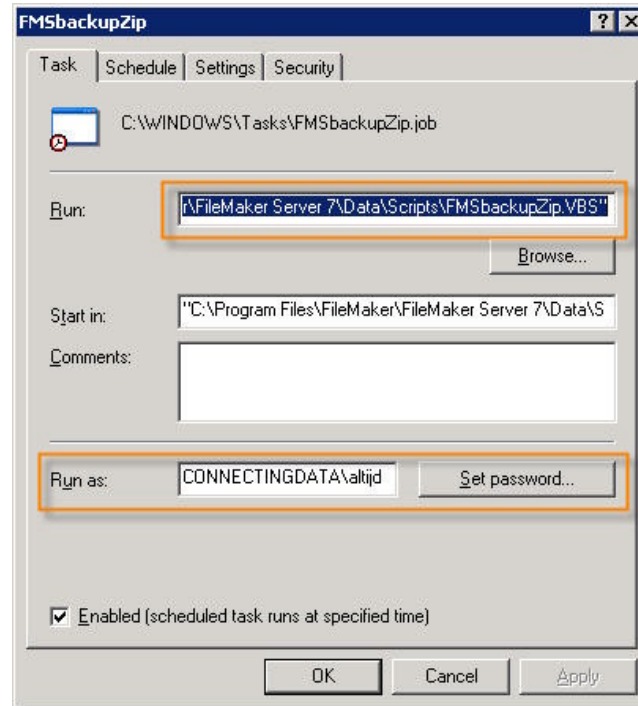


Figure 7 - The ZIP VBScript scheduled with the Windows Task Scheduler.

Note that when you use the Windows Task Scheduler you don't need to use the batch file workaround, Task Scheduler can run VBScripts natively.

To give you one more layer of protection, you can add one extra schedule to copy the zip backup set to another machine on the network. That's what line 16 in the VBScript is for. It saves the name of the current backup zip file in the registry (in the Local Machine section so it does not have to depend on anyone being logged in).

The 3rd schedule to copy the zip file to a network share runs a little after the second (after the zip creation is safely finished) and is also scheduled to run every hour (Figure 8). There's no reason why this couldn't be integrated in the first VBScript but using a separate schedule does spread the load on the server a bit. As you can see from Figure 8 we're using the Windows Task Scheduler instead of the FileMaker Server Scheduler. The FileMaker Server service running as "Local System" has no privileges across the network.

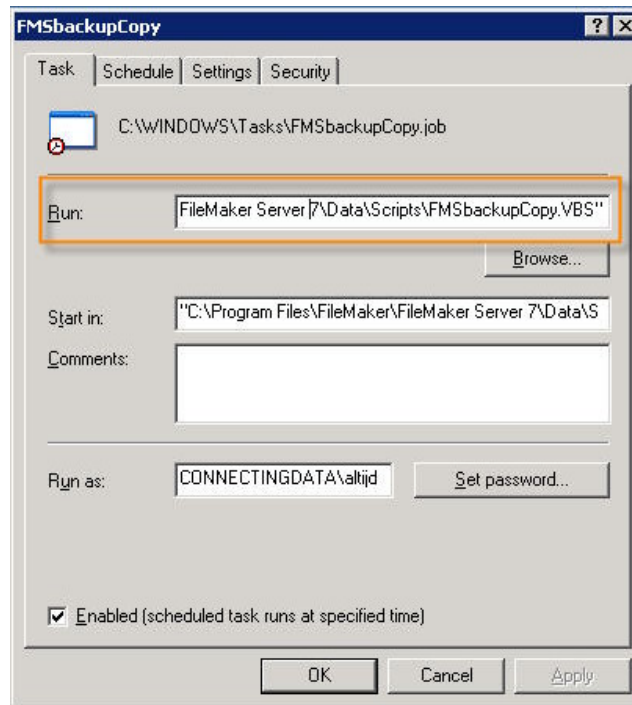


Figure 8 - 3rd schedule to run a VBscript that copies the date/time stamped zip file to a network share.

The VBscript is even simpler than the previous one and does this:

```

01
02 Set fso = CreateObject("scripting.filesystemobject")
03 Set SH = CreateObject("wscript.shell")
04
05 theKey = SH.RegRead("HKLM\SOFTWARE\FMS_backup\")
06 fso.CopyFile theKey, "\\p4\business\FMS_backups\"
07
08 Set fso = Nothing
09 Set SH = Nothing
10 WScript.Quit

```

Figure 9 - the VBscript that copies the zip file across the network.

Line 5 reads the name and location of the last backup zip file from the registry (where the first VBscript had saved it) and Line 6 copies the file to a network share.

While it is entirely possible to configure a FileMaker Server schedule to backup directly to a network share, I am not a fan of that. Backing up directly to a network share:

- is slow: it keeps the files in limbo much longer than a backup to the local hard disk;
- is fragile: if the network share is not available then no backup takes place which may leave you without a backup set to restore when things go wrong. If you're using a mapped drive letter to a network share then that adds to the vulnerability if that mapping is not there when the backup needs to happen.

What about Mac OSX?

The first backup schedule is the same on OSX: that's just FileMaker Server backing up to a local folder.

Below is an OSX shell script that does the equivalent of the second schedule. Download the file from: http://www.sumware.net/miscellaneous/backup_to_zip.sh.zip

```
#!/bin/bash
# 5 Dec 2005 Rob Russell SumWare Consulting
# only change the next two lines of this script to suit your installation

source="/Library/FileMaker Server 7/Data/Backups/hourly/"
destination="/Users/Shared/backups/fmbackups"

# source defines the folder you want to archive. Only archive
# FileMaker Server backups, not the master files hosted by FMServer.

# destination is the place you want the archives sent
# ( best this be a separate volume )

# suffix is the time stamp to append to the destination

suffix="$(date +%Y%m%d%H%M%S).zip"

# create symlink files in tmp folder, to deal with spaces in file paths

sourcelink="/tmp/sumware001.ln"

ln -s "$source" $sourcelink

destinationlink="/tmp/sumware002.ln"

ln -s "$destination$suffix" $destinationlink

# copy with pkzip compression and preserve resources

ditto -c -k --rsrc $sourcelink $destinationlink

# clean up the symlinks we created earlier

rm -f $sourcelink
rm -f $destinationlink
```

Save this shell script as a text file with a ".sh" extension and ensure that it is executable by FileMaker Server, which means it must be owned by fmserver and belong to fmserver group, or a group that fmserver belongs to. Alternatively, just make it read/write by everyone. You must also make it executable by the same owners and groups⁴. With this done you can schedule this shell script with the FileMaker Server Scheduler. If you want to move the backup set to another machine you can make use of the Secure Copy command line instruction "scp".

⁴ - The easiest way to do that is to use Batchmod or SuperGetInfo

Conclusion

The approach I've outlined in this article produces unlimited backup sets on both the local drive and on a network share. Network traffic is limited by only copying a zip archive of the files instead of the raw files. The setup is extremely simple and does not involve having to create a bunch of folders and schedules. That makes it easy to manage.

All feedback on this article is welcome: wim@connectingdata.com