

BackupAssist™ v9

Custom Scripting

User Guide

Contents

1. Overview	2
Scripting in BackupAssist.....	2
Pre and post backup scripts	2
Hints on scripting	2
2. Adding scripts to BackupAssist.....	3
The Manage Scripts screen.....	3
Variables.....	4
Reporting.....	4
3. Sample Scripts.....	5
Compression using 7za.....	5
Renaming a backup file.....	5
Copying a file	5
Restarting services.....	6
Retrying a network connection.....	6
4. Support and Resources	7
Contacting Technical Support	7
Pre backup variables	7
Post backup variables	7

1. Overview

Scripting in BackupAssist

Most users should be able to use BackupAssist without adding custom scripts. The software was designed to include all of the options needed to set up a best-practice backup strategy.

Before adding custom scripts, please check the options available in BackupAssist and the help files. It could be the case that the function you require is already available within the program. Where possible, you should use inbuilt options rather than custom scripts. If you think your requirement is common enough that it should be included in BackupAssist, please send an email explaining the requirement to support@backupassist.com.

We realize, however, that some of our users have requirements that we either did not foresee or, for concerns that they would detract from the software's ease of use, did not include. For those users we have provided the ability to add custom scripts to BackupAssist. Custom scripts can be used for a number of purposes: to copy, rename or compress backup files; to integrate with command line programs; to suspend and resume security or other programs during backup; and to restart services.

Pre and post backup scripts

There are four separate fields for adding scripts to BackupAssist:

- **Before each backup** - This script runs before any other step of the backup. You can add scripts here for authenticating to network resources, setting up backup folders or suspending programs that could interfere with the backup, such as anti-virus applications.
- **After each backup** - This script runs after all the other steps in the backup process have been completed. You can add scripts here to disconnect from network resources, restart any programs that were previously suspended, or to perform any number of post-backup operations.
- **After each successful backup** - This script runs before the *After each backup* script, but only if the backup completed successfully. You can enter commands here that you only want executed if the backup succeeded, such as copying the backup file to another location, or compressing the backup file.
- **After each failed backup** - This script runs just before the *After each backup* script, but only if the backup fails to complete. This is useful if you want to send a network broadcast alerting someone quickly after a failed backup, or to clean-up folders and quarantine failed backups.

Hints on scripting

- Any command that works within a .bat file will work in a BackupAssist script.
- Flow control, such as **if** and **goto** statements, will work in a BackupAssist script.
- BackupAssist includes a list of variables that can be used with its scripts. These variables are listed under the [Support and Resources](#) section at the end of this document.

2. Adding scripts to BackupAssist



Scripts can be added to a backup job after it has been created. The *Scripts* option is under the *Manage* menu on the Backup tab.

The Manage Scripts screen

To access the Scripts screen:

1. Select the BackupAssist, **Backup tab**.
2. Select **Manage** from the top menu. A list of all your backup jobs will be displayed.
3. Select the backup job that you want to add a script to.
4. Select **Edit** from the *Manage* menu.
5. Select **Scripts** from the pane on the left.
6. Enter your script into the appropriate field (e.g. Run after each backup).
Variables and report warnings can be added.
7. Once the script has been completed, select **Apply Changes**.

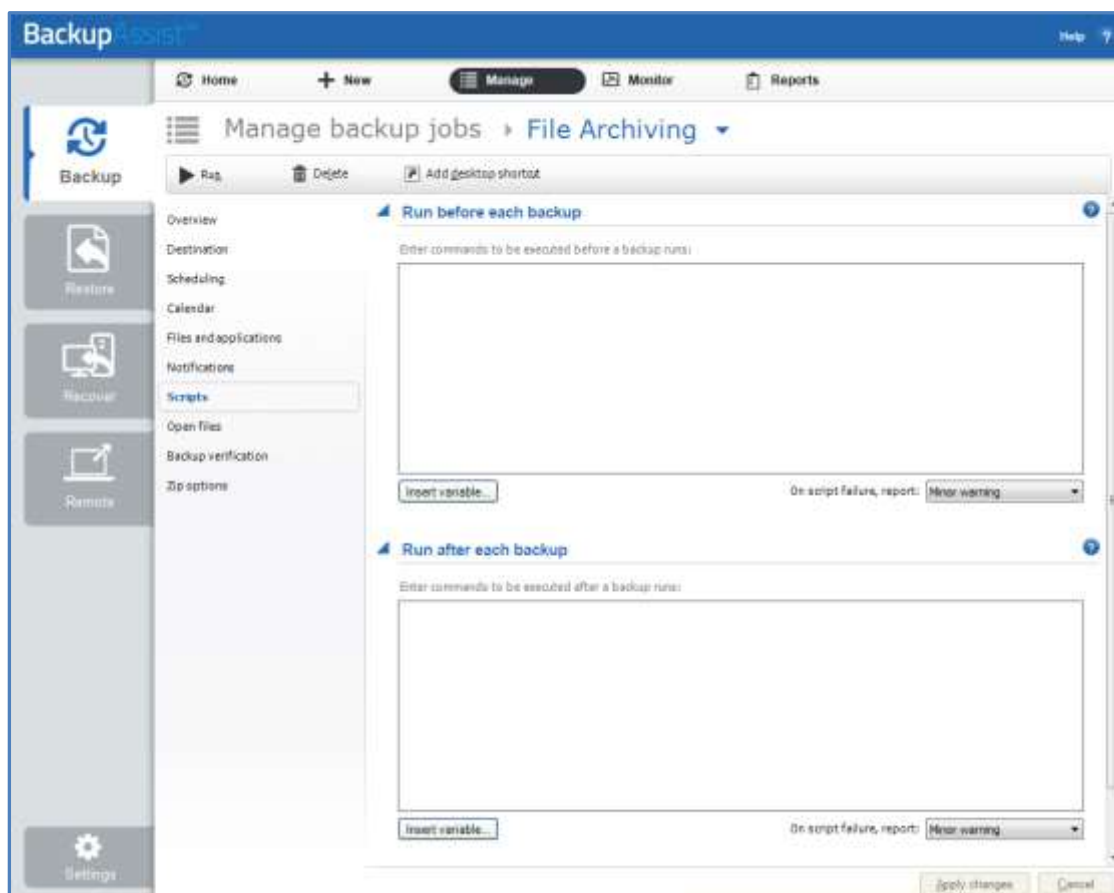


Figure 1: BackupAssist - Scripts screen

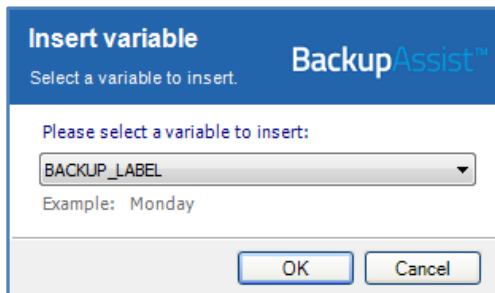
Important: Once a backup job has been created with scripting, it should be tested. To test your backup job, manually create a backup using the **Run** option in the *Manage* screen.

Variables

You can customize your script by inserting unique variables. A list of *Variable* descriptions can be found in the [Support and Resources](#) section of this document.

To add a variable:

- a. Click **Insert variable**.
- b. Choose a variable from the list.



- The variable will now be inserted into your script, enclosed with percentage signs.
An example of one variable: %JOB_NAME%.
- You can insert as many additional variables as required.
An example with two variables: net send Administrator "Backup %JOB_NAME% about to start to the %JOB_NAME% backup job "
This command will send a network broadcast to the Administrator user with the message (example): "Backup SQL Data about to start to the SQL Data backup job"

Reporting

You can determine how a failed script is flagged within the backup report by using the **On script failure report** drop-down menu. The options available are:

No error

If the script fails to execute the overall status of the backup job will not be affected.

Minor warning

If the script fails to execute a minor warning will be logged and then displayed in the backup report.

Major warning

If the script fails to execute a major warning will be logged and then displayed in the backup report.

Error

If the script fails to execute an error will be logged and then displayed in the backup report. The job will also terminate at this point before completing any further processes. This is useful for the *Before backup* script if you want to stop the backup job from running if the script fails.

3. Sample Scripts

Compression using 7za

To compress your backup file, type the following command in the *After each successful backup* section:

```
"[Install Directory]\7za.exe" a -y "%BACKUP_FILENAME%.zip" "%BACKUP_FILENAME%"
```

For example:

```
"C:\Program Files\BackupAssist\7za.exe" a -y "%BACKUP_FILENAME%.zip" "%BACKUP_FILENAME%"
```

'a' stands for archive, and -y answers 'yes' to any 7za prompt. Following this command you will most likely want to remove the original uncompressed file. This is achieved simply by entering:

```
del "%BACKUP_FILENAME%"
```

To add a password to the compressed file, use the -p argument like this:

```
"[Install Directory]\7za.exe" a -p[Your Password] -y "%BACKUP_FILENAME%.zip"  
"%BACKUP_FILENAME%"
```

Renaming a backup file

Please note that renaming backup files with a script may undermine the integrity of the rotation scheme you have chosen. For example, renaming Month1.bkf and Month2.bkf to the same file name will reduce the pool of backups you have available. If you still need to rename backup files, use the following command:

```
ren %BACKUP_FILENAME% [Your desired filename]
```

Make sure that your requested filename ends in .bkf if you are using an NTBackup job. You can, of course, use variables in your desired filename, for example:

```
ren %BACKUP_FILENAME% %WEEKDAY%%FILE_TIME%.bkf
```

This would result in filenames such as: Mon22.00.bkf, Tue22.00.bkf, etc.

Copying a file

Normally, if you need a copy of the backup file you can use the *Keep local copy* or *Create second backup copy* options found in the destination settings of your backup job. If you need an additional copy to be stored in another location, or you want to copy .pst, .tlog or any other files after the backup, you can use the xcopy command.

```
xcopy [Source] [Destination] /e /i /k /y
```

Restarting services

Sometimes restarting a service before a backup can help eliminate errors. Common services that benefit from restarting before backups are the *Removable Storage Manager* service and the *Volume Shadow Copy* service.

To restart the **Removable Storage Manager** service type the following into the 'before each backup' section:

```
net stop ntmssvc
net start ntmssvc
```

To restart the **Volume Shadow Copy** service:

```
net stop vss
net start vss
```

The names for other services can be found by going to the Windows Control Panel > Administrative Tools > Services, and right-clicking the service and selecting *Properties*.

Retrying a network connection

```
set count_=
set target_=!!!!
:LOOP
set count_=%count%!
net use \\192.168.1.1\users /user:[user] [pwd]
NETERRLEV=%ERRORLEVEL%
if %NETERRLEV% == 0 goto SUCCESS
if "%count_"=="%target%" goto ENDLOOP ping 24.24.24.24 -n 1 -w 5000
goto LOOP
:ENDLOOP
if %NETERRLEV% == 2 goto SOMETHING
if %NETERRLEV% == 1 goto NOPATH
goto END
:SUCCESS
Echo "the map was successful"
goto END
:NOPATH
echo "path was not found"
goto END
:SOMETHING
echo "something happened"
goto END
:END
```

4. Support and Resources

Contacting Technical Support

Should you have any questions regarding BackupAssist, please email support@backupassist.com and we will respond to you as soon as possible.

Similarly, if you have any suggestions for additional functionality in BackupAssist, or new products or add-ons, please also forward your feedback to support@backupassist.com

Pre backup variables

Scripts that are run before each backup can make use of any of the following variables:

BACKUP_FILENAME - the filename of the backup including the full path e.g. 'D:\Backups\Monday.bkf.

BACKUP_LABEL - the label that BackupAssist has given this backup; this does not necessarily match tape labels or filenames. E.g. 'Week 2', 'Monday' or 'Month 1'.

BACKUP_METHOD - a variable describing whether the backup's is overwrite or append.

BACKUP_SIZE - the size of the backup in bytes.

COMPUTER_NAME - the name of the computer performing the backup.

EJECT_AFTER_BACKUP - whether the media is to be ejected after the backup ('true' or 'false').

FREQUENCY_TYPE - whether this is a monthly, weekly, yearly, quarterly or daily backup.

IS_MEDIA_CHANGE - whether the media was to be changed before the backup.

JOB_NAME - the name of the backup job set in the 'Overview' section.

MEDIA_LABEL - same as the backup label.

RUN_LENGTH - the amount of time the backup took to run.

SCHEDULED_TIME - the time this backup job is scheduled to run.

SCHEDULED_DATE - the date this backup job is scheduled to run.

STATUS - whether the backup succeeded or failed.

START_TIME - the time the backup starts.

START_DATE - the date the backup starts .

You can specify the date or time format by adding :{} , eg. %START_TIME:{HH:mm}.

The string inside { } can be either a standard date/time format string or a customized date/time format

- Standard date/time format: [https://msdn.microsoft.com/en-us/library/az4se3k1\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/az4se3k1(v=vs.110).aspx)
- Customized date/time format: [https://msdn.microsoft.com/en-us/library/8kb3ddd4\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/8kb3ddd4(v=vs.110).aspx)

Post backup variables

In addition, scripts that are run after each backup can make use of the following variables:

- **BACKUP_SIZE** - the size of the backup in bytes.
- **RUN_LENGTH** - the amount of time the backup took to run.
- **STATUS** - whether the backup succeeded or failed.