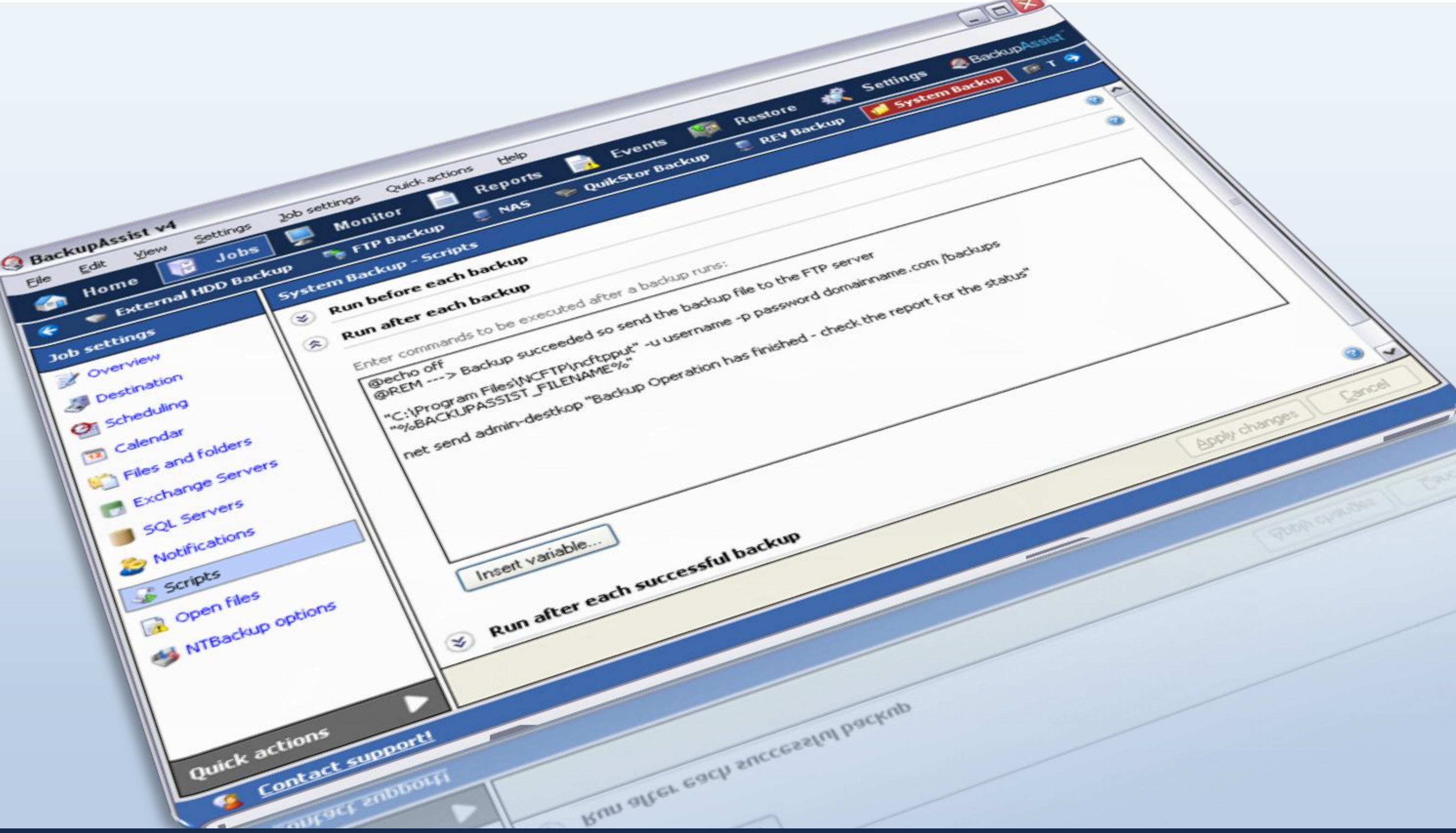

Enter commands to be executed after a backup runs:

```
@echo off
@REM ---> Backup succeeded so send the backup file to the FTP server
"C:\Program Files\NCFTP\ncftpput" -u username -p password domainname.com /backups "%BACKUPASSIST_FILENAME%"

net send admin-destkop "Backup Operation has finished - check the report for the status"
```

Custom Scripting

BackupAssist™
Windows® Backup Made Easy!

WHITEPAPER

BackupAssist Version 4

www.BackupAssist.com

Cortex I.T.

1. Introduction and Overview

1.1 Scripting in BackupAssist

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

Before adding custom scripts, please check the options in the console and the help files. It could be the case that the function you require is already available elsewhere in 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 in the program, please send us an email at 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.

1.2 Pre and post backup scripts

There are four separate sections 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 to a failed backup, or to clean-up folders and quarantine failed backups.

1.3 Hints on scripting

Any command that works within a .bat file will work in a BackupAssist script. This includes flow control (if, goto statements, etc). Section 2.5 demonstrates the use of flow control.

A list of variables included in BackupAssist that can be used within scripts can be found in section 3.

2. Sample Scripts

2.1 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%”
```

2.2 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.

2.3 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
```

2.4 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’.

2.5 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
```

3. Reference

3.1 Pre backup variables

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

- **COMPUTER_NAME** - the name of the computer performing the backup.
- **JOB_NAME** - the name of the backup job set in the ‘Overview’ section.
- **TIME** - the current time, e.g. ‘11:30:00 AM’.
- **DAY** - the day of the month, e.g. ‘12’ for the twelfth day of the month.
- **MONTH** - the month of the year, e.g. ‘5’ for May.
- **YEAR** - the year e.g. ‘2007’
- **DATE** - the date using your system’s date format, e.g. ‘08/24/2007’ for America.
- **FILE_DATE** - a file friendly year - month - day format for the date, e.g. ‘2007-08-24’.
- **FILE_TIME** - a file friendly 24 hour time format e.g. ‘15.09’ for 9 minutes past 3pm.
- **WEEKDAY** - a short description of the current day of the week e.g. ‘Mon’

- **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.
- **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.
- **EJECT_AFTER_BACKUP** - whether the media is to be ejected after the backup (‘true’ or ‘false’).
- **MEDIA_LABEL** - same as the backup label.
- **BACKUP_FILENAME** - the filename of the backup including the full path e.g. ‘D:\Backups\Monday.bkf’

3.2 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.