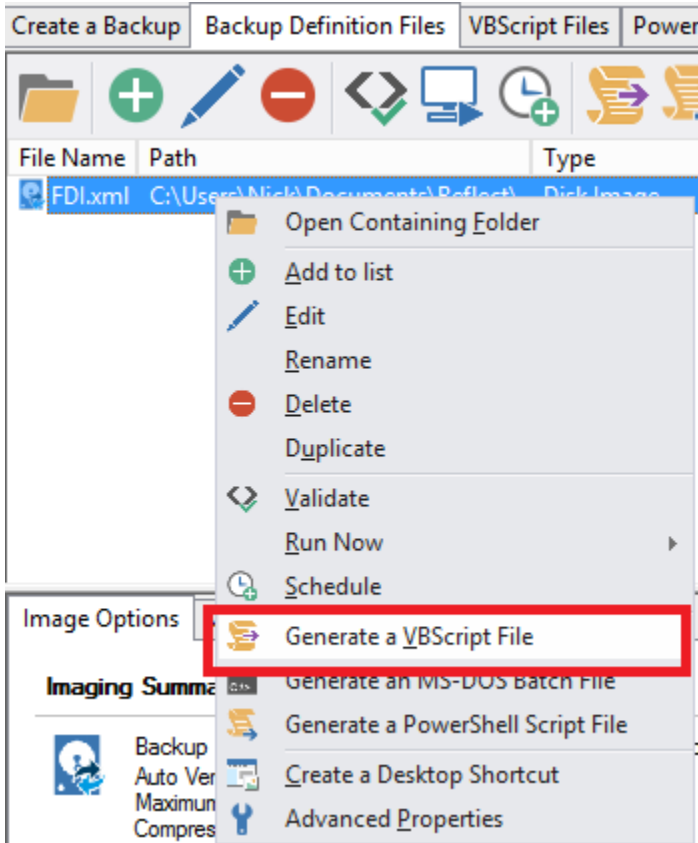


Generating a VBScript source file

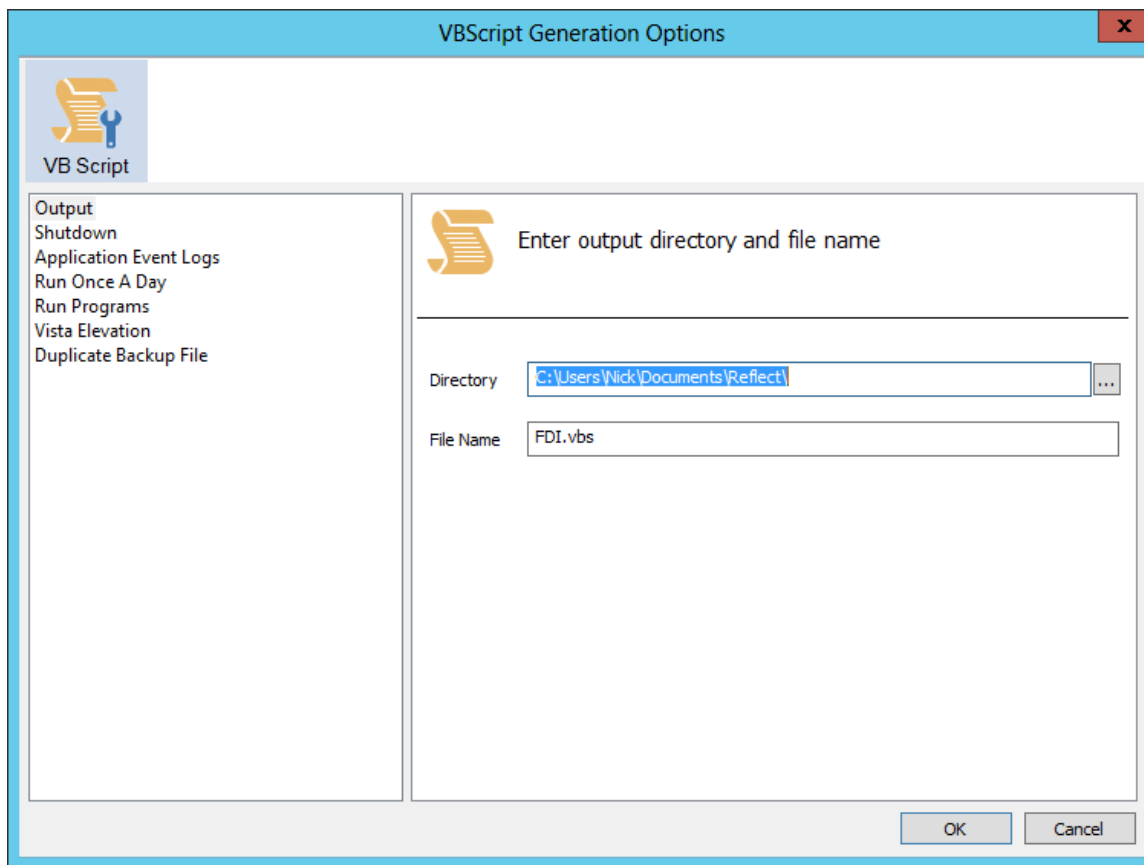
For an example of a generated VBScript source file please see [Example VBScript Source](#)

To generate a VBScript file select the 'XML Definition Files' tab on the main window then right click on a file and select '**Generate a VBScript file**'.



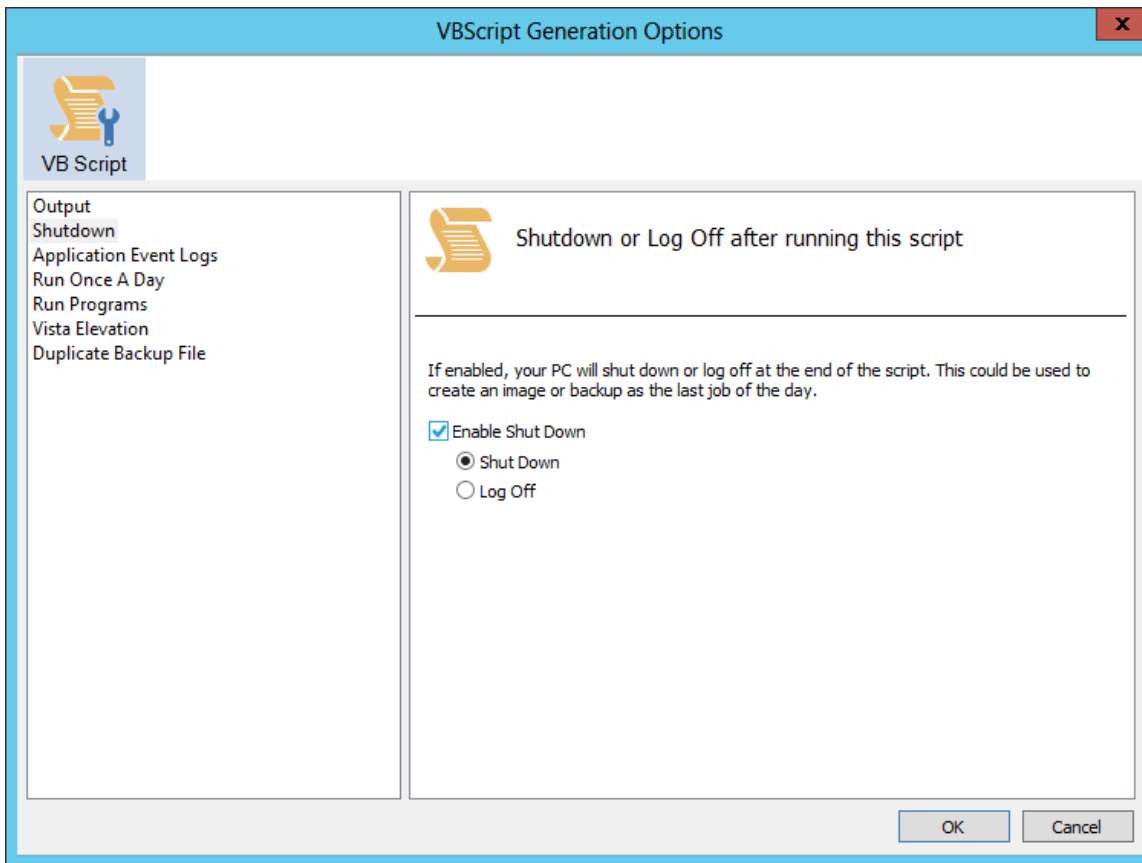
This opens the **VBScript Generation Options** dialog:

Output



Property	Value
Directory	The folder where the source file is saved
File name	The name of the VBscript source file. This defaults to the XML file name with a .vbs extension

Shutdown



' Issue shutdown command

```
objShell.Run "shutdown -s"
```

Property	Value
Enable Shut Down	Select to enable this property
Shut down	The PC will shut own at the end of the backup
Log Off	The user will be logged out at the end of the backup

Application Event logs

Please note that Macrium Reflect Server and Server Plus Edition includes Windows Event logging built-in.

VBScript Generation Options



VB Script

- Output
- Shutdown
- Application Event Logs
- Run Once A Day
- Run Programs
- Vista Elevation
- Duplicate Backup File



Write messages to the Reflect log in this script

☒ Write a message on success

Macrium Reflect - Successful Backup

☒ Write a message on failure

Macrium Reflect - Backup Failed

```

if iReturnCode = 2 then
' Handle XML validation error
elseif iReturnCode = 1 then
' Handle backup error
    objShell.LogEvent 1, "Macrium Reflect - VBScript Backup Failed"
elseif iReturnCode = 0 then
' Everything OK
    objShell.LogEvent 0, "Macrium Reflect - Successful VBScript Backup"
end if

```

Property	Value
Write a message on success	Enable a Windows Event on successful backup
	Enter the message text in the edit box below
Write a message on failure	Enable a Windows Event for failed backups
	Enter the message text in the edit box below

Note: The Windows Event logs generated by this script will have WSH (Windows Scripting host) as the message source

Level	Date and Time	Source	Task Category
 Information	07/04/2015 14:15:35	WSH	None

Event 0, WSH

General		Details	
Macrium Reflect - Successful VBScript Backup			
Log Name:	Application	Logged:	07/04/2015 14:15:35
Source:	WSH	Task Category:	None
Event ID:	0	Keywords:	Classic
Level:	Information	Computer:	Office2
User:	N/A		

Run Once a Day

VBScript Generation Options

 VB Script

Output
Shutdown
Application Event Logs
Run Once A Day
Run Programs
Vista Elevation
Duplicate Backup File

 Only run this script once a day

It can be useful to execute a script the first time you logon, logoff or shut down. Using this component you can be sure that an event triggered task only runs this script once a day.

☐ Run Once a Day.

OK Cancel

```

' The following function call ensures that this script only runs once a day
If HasRunToday Then
    WScript.Quit
End If

.....

*****
'* Function: HasRunToday
'*
'* Purpose:  determines if this script has run today
'*
'*
'*
'* Input:    None
'* Output:   true if has run today false otherwise
'*
*****
Function HasRunToday
    Dim RegScriptKey
    Dim LastRunDate
    Dim objShell

    Set objShell = WScript.CreateObject("WScript.Shell")
    RegScriptKey = "HKCU\SOFTWARE\Macrium\Reflect\Scripts\" & WScript.ScriptFullName & "\LastRun"
    'Check if script has run today
    ON ERROR RESUME NEXT
    LastRunDate = objShell.RegRead(RegScriptKey)
    If LastRunDate = cstr(Date) Then
        HasRunToday = true
    Else
        objShell.RegWrite RegScriptKey, Date,"REG_SZ"
        HasRunToday = false
    End If
    Set objShell = nothing
End Function

```

Property	Value
Run Once a Day	If selected, will only enable this script to run once per day. This is useful for example if you want a backup to happen at first login or shutdown.

Run Programs

VBScript Generation Options

VB Script

- Output
- Shutdown
- Application Event Logs
- Run Once A Day
- Run Programs
- Vista Elevation
- Duplicate Backup File

Run programs or scripts from within this script

☒ Run a program or script at the start.

File name: ...

Parameters:

☒ Run a program or script at the end.

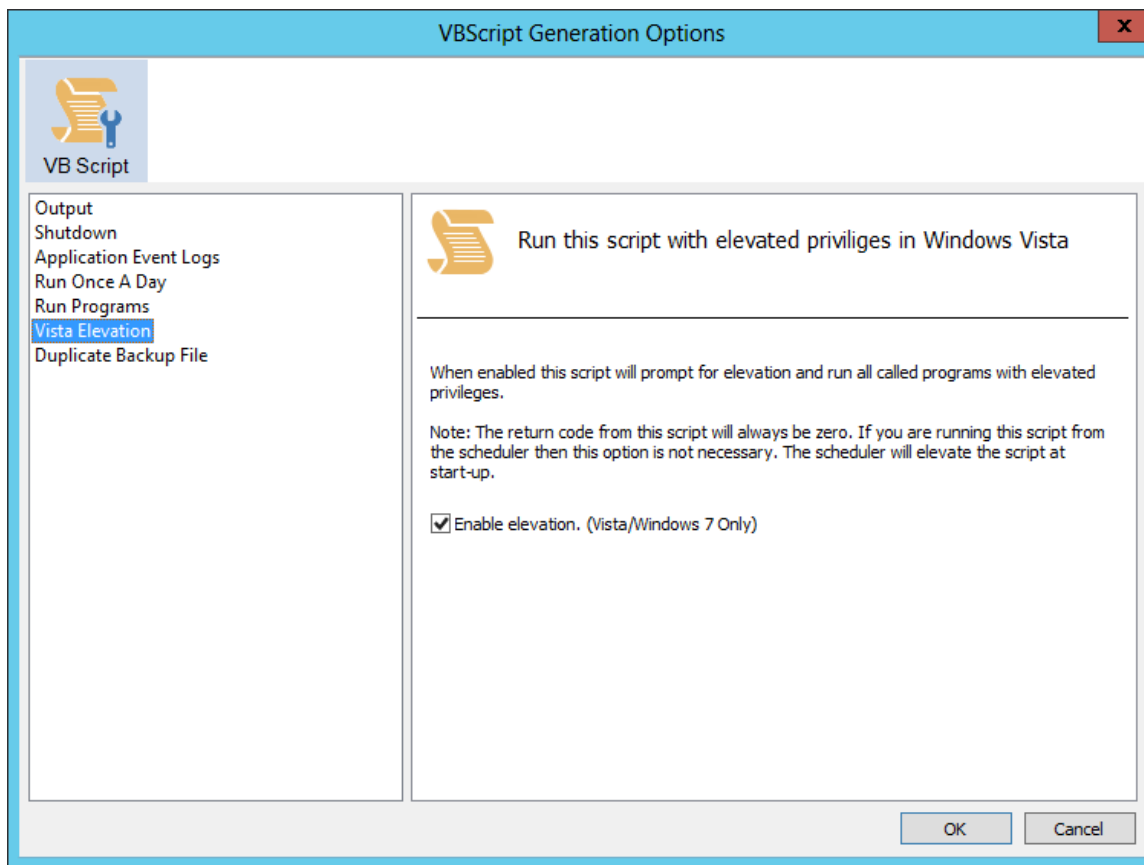
File name: ...

Parameters:

```
' Run program before backup
objShell.Run ""C:\Users\Public\Admin\Before backup.bat"", 1, true
' Do the backup
ExitCode = Backup ("""C:\Program Files\Macrium\Reflect\reflect.exe"" -e -w <BACKUP_TYPE> ""C:
\Users\Dev\Documents\Reflect\example.xml""")
' Run program after backup
objShell.Run ""C:\users\public\Admin\After backup.exe"", 1, true
```

Property	Value
Run a program or script at the start	Select to enable a program or script to run before the backup starts
File name	The path and executable file name
Parameters	Optional command line parameters for the program
Run a program or script at the start	Select to enable a program or script to run when the backup ends
File name	The path and executable file name
Parameters	Optional command line parameters for the program

Vista Elevation



```

Sub VBMain()
    Dim objShell
    Dim ExitCode
    ' Elevate this script for Admin privileges in Vista
    Elevate

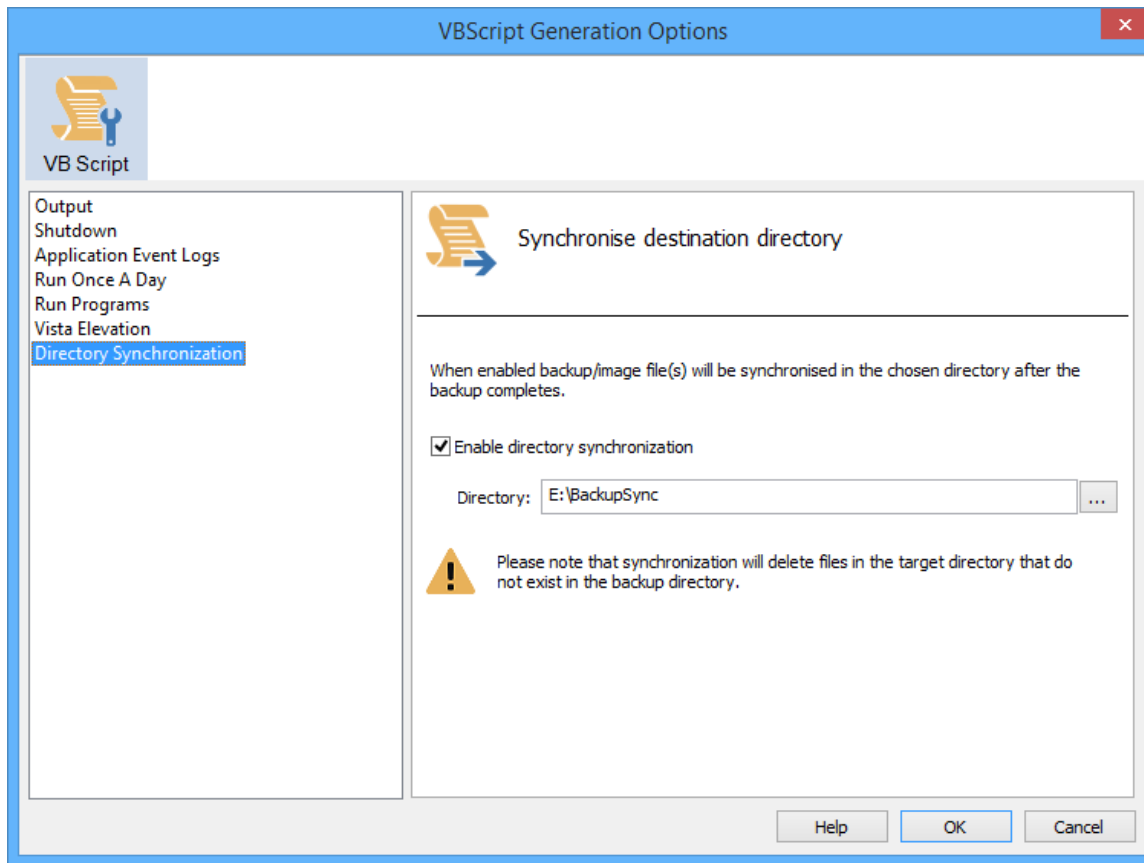
    .....

    *****
    '* Sub:      Elevate
    '*
    '* Purpose: Elevates this script for Vista UAC.
    '*          This means that only one UAC Elevation prompt is displayed and
    '*          functions/programs will not fail if they require admin privileges
    '*
    '* Input:   None
    '* Output:  None
    '*
    *****
Sub Elevate
    Dim ShellApp
    Dim objShell
    Dim objWshProcessEnv
    ' Don't elevate if run from Windows Task Scheduler
    If WScript.Arguments.Count > 0 Then
        If WScript.Arguments.Item(0) = "-s" then
            Exit Sub
        End If
    End If
    Set objShell = WScript.CreateObject("WScript.Shell")
    Set objWshProcessEnv = objShell.Environment("PROCESS")
    If objWshProcessEnv("ELEVATED_APP") <> "True" Then
        objWshProcessEnv("ELEVATED_APP") = "True"
        Set ShellApp = CreateObject("Shell.Application")
        Call ShellApp.ShellExecute(""" & WScript.FullName & """, """" & WScript.ScriptFullName & """" &
        " " & GetBackupTypeParameter, , "runas")
        set ShellApp = nothing
        Set objWshProcessEnv = nothing
        wscript.quit
    End If
    Set objWshProcessEnv = nothing
    Set objShell = nothing
End Sub

```

Property	Value
Enable Elevation	If selected will enable UAC elevation for the entire script. This enables functions and programs to run outside the context of Macrium Reflect without requesting further elevation.

Directory Synchronization



SynchroniseDirectories ExitCode, "C:\C++"

```
.....

*****
!* Function: RobocopyExists
!*
!* Purpose: determines whether the program Robocopy can be called from the
!*           command line.
!*
!* Input:    None
!* Output:   boolean - true if Robocopy exists
!*
*****

Function RobocopyExists()
    On Error Resume Next
    Dim objShell
    Dim objExec
    Set objShell = WScript.CreateObject("WScript.Shell")
    Set objExec = objShell.Exec ("robocopy.exe", 0)

    If Err.Number = 0 Then
        RobocopyExists = True
    Else
        RobocopyExists = False
    End If
End Function

*****
!* Function: SynchroniseDirectories
!*
!* Purpose: Copies all files created by the previous backup / image
!*           to a supplied directory. Uses Macrium environment variables to
!*           determine which files to copy.
!*
!* Input:   ExitCode - The exit code of the last backup
!*           strBackupDirectory - Directory to copy to
```

```

' *
' *****
Function SynchroniseDirectories(Byval ExitCode, Byval strSyncDirectory)
    Dim objShell
    Dim objWshProcessEnv
    Dim strEnvPrefix
    Dim strBackupDirectory
    Dim strCmdLine
    Dim iReturnCode
    Dim fs
    Dim objSyncFiles
    Dim objBackupDirectory
    Dim objSyncDirectory
    Dim objBackupFile
    Dim objSyncFile
    Dim strExtension
    Dim dateBackupFile
    Dim dateSyncFile

' Only copy files if backup was successful
    if ExitCode <> 0 Then
        Exit Function
    End If

    Set objShell = WScript.CreateObject("WScript.Shell")
    Set objWshProcessEnv = objShell.Environment("VOLATILE")

' Get the prefix for the last backup set
    strEnvPrefix = objWshProcessEnv("MACRIUM_PREFIX")

' Get the directory where we just created a backup
    strBackupDirectory = objWshProcessEnv(strEnvPrefix + "_DIRECTORY")

    If right(strBackupDirectory, 1) = "\" Then strBackupDirectory = left(strBackupDirectory, len
(strBackupDirectory)-1)
    If right(strSyncDirectory, 1) = "\" Then strSyncDirectory = left(strSyncDirectory, len
(strSyncDirectory)-1)

    If RobocopyExists Then
        strCmdLine = "robocopy ""<SOURCE>"" ""<DESTINATION>"" *.mr* /copy:DAT /lev:0 /purge /r:0"
    Else
' Robocopy does not exist - using xcopy
        Set fs = CreateObject("Scripting.FileSystemObject")
        Set objBackupDirectory = fs.GetFolder(strBackupDirectory)
        Set objSyncDirectory = fs.GetFolder(strSyncDirectory)

        For Each objSyncFile in objSyncDirectory.Files
            strExtension = fs.GetExtensionName(objSyncFile)
            dateSyncFile = objSyncFile.DateLastModified
            If Left(strExtension, 2) = "mr" Then
' Check if file has been deleted
                If Not(fs.FileExists(strBackupDirectory+"\\"+objSyncFile.name)) Then
                    fs.DeleteFile(objSyncFile)
                Else
' Check if file has been modified
                    Set objBackupFile = fs.GetFile(strBackupDirectory+"\\"+objSyncFile.name)
                    dateBackupFile = objBackupFile.DateLastModified
                    If DateDiff("m", dateBackupFile, dateSyncFile) <> 0 Then
                        fs.DeleteFile(objSyncFile)
                    End If
                End If
            End If
        Next

        strCmdLine = "xcopy ""<SOURCE>\*.mr*"" ""<DESTINATION>"" /c /d /h /i /v /y"
    End If

    strCmdLine = Replace(strCmdLine, "<SOURCE>", strBackupDirectory)
    strCmdLine = Replace(strCmdLine, "<DESTINATION>", strSyncDirectory)
    iReturnCode = objShell.Run(strCmdLine, 0, true)

    if iReturnCode <> 0 then
' Handle synchronisation error
        else
' Everything OK

```

```
        end if  
  
    ' Clean up  
        Set objShell      = nothing  
        Set objWshProcessEnv = nothing  
End Function
```

Property	Value
Enable directory synchronization	If enabled will use the MS utility RoboCopy to synchronize the backup target directory with the supplied directory.
Directory	Enter the folder that you want to copy/archive your backup files to

Click **'OK'** to generate the VBScript source file
