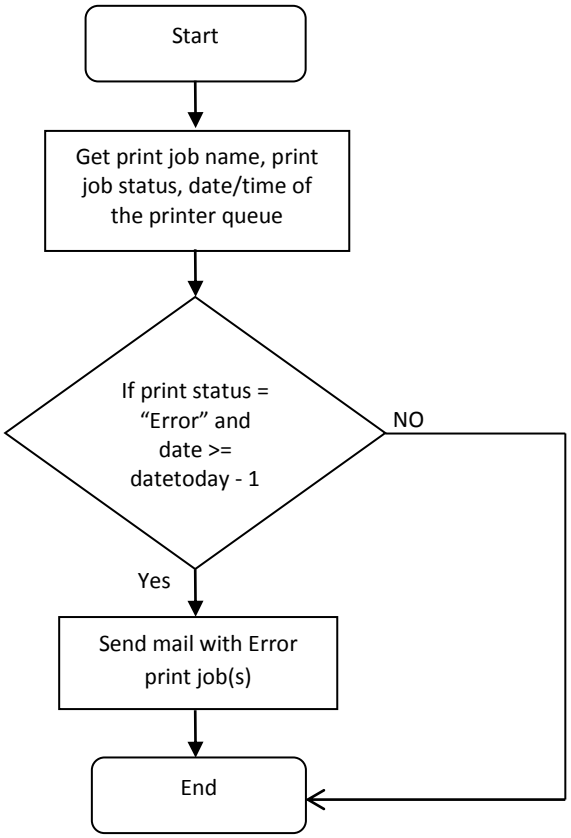
Monitoring of Print jobs with Emailing functionality

1.0 Introduction

The script has been designed and implemented on Visual Basic script; the basic task behind this tool is to monitor the print jobs status on a defined printer. On errors, it will send out a notification via mail.

2.0 Flowchart



2.0 Script

```
‘Power-shell 1.0 Script written by: Arvind Ramtohul

‘Date written: March 2012

‘ ----------------------------------------------------------------------------------------------------------------

‘ Initializing global variables


StrServer = "servername"            ‘Used in email function and main script

StrPrinter = "printername"          ‘Printer name variable

Dim BodyTxt                         ‘The body of the mail

Dim BodyLine                        ‘The variable that will contain the error line

Dim Error = 0                       ‘Initial error variable set to 0

‘ ----------------------------------------------------------------------------------------------------------------

Set DateTime = CreateObject("WbemScripting.SWbemDateTime")

‘Retrieving the WMI time

Set objWMIService = GetObject("winmgmts:" _

& "{impersonationLevel=impersonate}!\\" & strServer & "\root\cimv2")

‘Retrieving the WMI information of the machine name


BodyTxt= "Print Job" & "                    " & "Time Submitted" & vbcrlf & string
(117,"_")

‘ ----------------------------------------------------------------------------------------------------------------

DateToday = DateAdd("d",-1 * 1,now)

‘Using date yesterday to search the errors

Set colInstalledPrinters =  objWMIService.ExecQuery _

("Select * from Win32_PrintJob Where Name like '%"   StrPrinter “%' and (JobStatus like
'%Error%' ) ")

‘Extracting the print jobs which ended in error
```

```
For Each colPrintJobs in colInstalledPrinters
'Run through all the print jobs that ended in error
        DateTime.Value = colPrintJobs.TimeSubmitted
        dtmActualTime = DateTime.GetVarDate(USE_LOCAL_TIME)
        'Converting the WMI time into local system time
        If  dtmActualTime >=DateToday Then
                Error = 1
                BodyLine = colPrintJobs.Document & "      " & dtmActualTime
                'Bodyline contains the "error" jobs with its respective failing date and time
        End if
BodyTxt = BodyTxt & vbcrlf & BodyLine
Next

If Error = 1 Then
        'At error, script will send out a notification
        BodyLine = "Above job(s) have ended in error. Please check Printer!"
        BodyTxt = BodyTxt & vbcrlf & vbcrlf & BodyLine
        Check = "Error"
        Sendmail BodyTxt, Check, StrPrinter, StrServer
End if


Function SendMail(txtbody, Check, StrPrinter, STrServer)
dim objMsg
dim msweb
msweb = "http://schemas.microsoft.com/cdo/configuration/"
set objMsg = CreateObject("CDO.Message")
```

```
objMsg.Subject          = "PRINTER " & StrPrinter & " - North PD " & StrServer & " @ " & Trim(cstr(now)) &_

" (" & trim(cstr(Check)) & " )"

objMsg.From             = " sourceaddress@mail.com "

objMsg.To               = "destinationaddress1@mail.com, destinationaddress2@mail.com , destinationaddressZZZ@mail.com"

objMsg.cc               = " destinationaddress3@mail.com "

objMsg.TextBody         = txtbody


'This section provides the configuration information for the remote SMTP server.

'Normally you will only change the server name or IP.

'Server port (typically 25)

'Use SSL for the connection (False or True)

'Connection Timeout in seconds1 (the maximum time CDO trying to connect to the SMTP server)

objMsg.Configuration.Fields.Item(msweb & "sendusing").Value = 2

objMsg.Configuration.Fields.Item(msweb & "smtpserver").Value = "smtp.client.com"

objMsg.Configuration.Fields.Item(msweb & "smtpserverport").Value = 25

objMsg.Configuration.Fields.Item(msweb & "smtpusessl").Value = False

objMsg.Configuration.Fields.Item (msweb & "smtpconnectiontimeout").Value = 60

objMsg.Configuration.Fields.Update

'========== End remote SMTP server configuration section==


objMsg.Send

Set objMsg =nothing

End function
```