XII. Annexe 3 : de Vbs à Powershell, documentation adaptée d'un document Microsoft

VBScript Function	Windows PowerShell Equivalent
Abs	$a = [math]::abs(-15)
Array	$a = "red","orange","yellow","green","blue","indigo","violet"
Asc	$a = [byte][char] "A"
Atn	$a = [math]::atan(90)
CBool	$a = 0 $a = [bool] $a
CByte	$a = "11.45" $a = [byte] $a
CCur	$a = "{0:C}" -f 13
CDate	$a = "11/1/2006" $a = [datetime] $a
CDbl	$a = "11.45" $a = [double] $a
Chr	$a = [char]34
CInt	$a = "11.57" $a = [int] $a
CLng	$a = "123456789.45" $a = [long] $a
Cos	$a = [math]::cos(45)
CreateObject	$a.visible = $True $a = new-object -comobject Excel.Application -strict
CSng	$a = "11.45" $a = [single] $a
CStr	$a = 17 $a = [string] $a
Date	$a = get-date –format d
DateAdd	$a = (get-date).AddDays(37) (get-date).AddHours(37) (get-date).AddMilliseconds(37) (get-date).AddMinutes(37) (get-date).AddMonths(37) (get-date).AddSeconds(37) (get-date).AddTicks(37) (get-date).AddYears(37) $a = ((get-date).AddHours(2)).AddMinutes(34)
DateDiff	$a = New-TimeSpan $(Get-Date) $(Get-Date –month 12 -day 31 -year 2006 -hour 23 -minute 30) $a.Days Days : 109 Hours : 3 Minutes : 55 Seconds : 0 Milliseconds : 0 Ticks : 94317000000000 TotalDays : 109.163194444444 TotalHours : 2619.91666666667 TotalMinutes : 157195 TotalSeconds : 9431700 TotalMilliseconds : 9431700000
DatePart	$a = (get-date).day

	$a = (get-date).dayofweek $a = (get-date).dayofyear $a = (get-date).hour $a = (get-date).millisecond $a = (get-date).minute $a = (get-date).month $a = (get-date).second $a = (get-date).timeofday $a = (get-date).year $a = (get-date).hour
DateSerial	MyDate1 = DateSerial(2006, 12, 31) $a = get-date -y 2006 -mo 12 -day 31
DateValue	$a = [datetime] "12/1/2006"
Day	$a = (get-date).day
Eval	$a = 2 + 2 -eq 45
Exp	$a = [math]::exp(2)
Filter	$a = "Monday","Month","Merry","Mansion","Modest" $b = ($a | where-object {$_ -like "Mon*"})
FormatCurrency	$a = 1000 $a = "{0:C}" -f $a
FormatDateTime	$a = (get-date).tolongdatestring() $a = (get-date).toshortdatestring() $a = (get-date).tolongtimestring() $a = (get-date).toshorttimestring()
FormatNumber	$a = 11 $a = "{0:N6}" -f $a
FormatPercent	$a = .113 $a = "{0:P1}" -f $a
GetLocale	$a = (get-culture).lcid $a = (get-culture).displayname
Hex	$a = 4517 $a = "{0:X}" -f $a
Hour	$a = (get-date).hour
InputBox	$a = new-object -comobject MSScriptControl.ScriptControl $a.language = "vbscript" $a.addcode("function getInput() getInput = inputbox(`"Message box prompt`",`"Message Box Title`") end function") $b = $a.eval("getInput")
InStr	$a = "wombat" $b = $a.contains("m") $b = $a.indexof("m")
InStrRev	$a = "1234x6789x1234" $b = $a.lastindexofany("x")
Int/Fix	$a = 11.98 $a = [math]::truncate($a)
IsArray	$a = 22,5,10,8,12,9,80 $b = $a -is [array]
IsDate	$a = 11/2/2006 $a -is [datetime] $a = [datetime] "11/2/2006"
IsEmpty	$a = "" $b = $a.length -eq 0
IsNull	$a = $z -eq $null
IsNumeric	$a = 44.5 [reflection.assembly]::LoadWithPartialName("'Microsoft.VisualBasic")

	$b = [Microsoft.VisualBasic.Information]::isnumeric($a)
IsObject	$a = new-object -comobject scripting.filesystemobject $b = $a -is [object]
Join	$a = "h","e","l","l","o" $b = [string]::join("", $a)
LBound	$a = 1,2,3,4,5,6,7,8,9 $b = $a.getlowerbound(0)
LCase	$a = "ABCDEFGHIJKLMNOPQRSTUVWXYZ" $a = $a.ToLower()
Left	$a="ABCDEFGHIJKLMNOPQRSTUVWXYZ" $a = $a.substring(0,3)
Len	$a = "abcdefghijklmnopqrstuvwxyz" $b = $a.length
Log	$a = [math]::log(100)
LTrim	$a = "..........123456789.........." $a = $a.TrimStart()
RTrim	$a = "..........123456789.........." $a = $a.TrimEnd()
Trim	$a = "..........123456789.........." $a = $a.Trim()
Mid	$a="ABCDEFG" $a = $a.substring(2,3)
Minute	$a =(get-date).minute
Month	$a = get-date -f "MM" $a = [int] (get-date -f "MM")
MonthName	$a = get-date -f "MMMM"
MsgBox	$a = new-object -comobject wscript.shell $b = $a.popup("This is a test",0,"Test Message Box",1)
Now	$a = get-date
Oct	$a = [Convert]::ToString(999,8)
Replace	$a = "bxnxnx" $a = $a -replace("x","a")
RGB	$blue = 10 $green= 10 $red = 10 $a = [long] ($blue + ($green * 256) + ($red * 65536))
Right	$a = "ABCDEFGHIJKLMNOPQRSTUVWXYZ" $a = $a.substring($a.length - 9, 9)
Rnd	$a = new-object random $b = $a.next(1,100) $b = $a.next()
Round	$a = [math]::round(45.987654321, 2)
ScriptEngine	$a = (get-host).version
ScriptEngineBuildVersion	$a = (get-host).version.build
ScriptEngineMajorVersion	$a = (get-host).version.major
ScriptEngineMinorVersion	$a = (get-host).version.minor
Second	$a = (get-date).second
Sgn	$a = [math]::sign(-453)
Sin	$a = [math]::sin(45)
Space	$a = " " * 25 $a = $a + "x"
Split	$a = "atl-ws-01,atl-ws-02,atl-ws-03,atl-ws-04" $b = $a.split(",")

Sqr	$a = [math]::sqrt(144)
StrComp	$a = "dog" $b = "DOG" $c = [String]::Compare($a,$b,$True)
String	$a = "=" * 20
StrReverse	$a = "Scripting Guys" for ($i = $a.length - 1; $i -ge 0; $i--) {$b = $b + ($a.substring($i,1))}
Tan	$a = [math]::tan(45)
Time	$a = get-date -displayhint time
TimeSerial	$a = get-date -h 17 -mi 10 -s 45 -displayhint time
TimeValue	$a = [datetime] "1:45 AM"
TypeName	$a = 55.86768 $b = $a.gettype().name
UBound	$a = "a","b","c","d","e" $a.getupperbound(0) $a.length-1
UCase	$a = "abcdefghijklmnopqrstuvwxyz" $a = $a.ToUpper()
WeekdayName	$a = (get-date).dayofweek $a = (get-date "12/25/2007").dayofweek
Year	$a = (get-date).year $a = (get-date "9/15/2005").year
Const Statement	set-variable -name ForReading -value 1 -option constant
Dim Statement	$a = [string]
Execute Statement	$a = "get-date" invoke-expression $a
Function Statement	function multiplynumbers { $args[0] * $args[1] } multiplynumbers 38 99
On Error Statement	$erroractionpreference = "SilentlyContinue" Incidentally, your choices for this variable include: SilentlyContinue Continue (the default value) Inquire Stop
Option Explicit Statement	set-psdebug –strict set-psdebug -off
Private Statement	$Private:a = 5
Public Statement	$Global:a = 199
Randomize Statement	$a = new-object random $b = $a.next()
ReDim Statement	$a = 1,2,3,4,5 $a = $a + 100 $a = $a[0..2]
Set Statement	$a = new-object -comobject Excel.Application $a.visible = $True
Stop Statement	set-psdebug –step set-psdebug -off
Sub Statement	function multiplynumbers { $args[0] * $args[1] } multiplynumbers 38 99
Description Property	$a = $error[0].ToString()
HelpContext Property	$a = $error[0].helplink
HelpFile Property	$a = $error[0].helplink
Number Property	ScriptHalted $error[0].errorrecord
Source Property	$a = $error[0].source

Clear Method	$error[0] = "" $error.clear()
Raise Method	$b = "The file could not be found."; throw $b