

Récupérer la clé produit Microsoft Windows

Un petit script permet de récupérer les informations concernant votre licence.

Copier ce code dans un fichier **viewpk.wsf**. L'extension du fichier est importante.

[viewpk.wsf](#)

```
<job>
<runtime>
<description  author="jcb">
-----
Ce script en ligne de commande détermine, à partir des clefs
"...\\...\\DigitalProductID", la valeur de la "ProductKey"
(clef de produit) sous la forme 5x5 :
        xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
Chaque caractère fait partie de l'ensemble BCDFGHJKMPQRTVWXY2346789
Par défaut il examine le "DigitalProductID" de Windows, mais il est
capable de trouver et analyser ces clefs pour tous les produits
Microsoft.
(Office, FrontPage, Visio, ...)
Ce script affiche aussi la "RAWkey" correspondante (valeur binaire,
exprimée en hexadécimal, en notations "little endian" et "big endian")
Inversement, il sait aussi convertir une RAWKey en ProductKey.

Enfin il sert à calculer (partiellement) le ProductID à partir
de la ProductKey ou de la RAWkey.

JCB © 2005
-----
</description>
<named name = "c"
    helpstring = "Nom NetBIOS d'ordinateur
    Si ce paramètre est absent, on retient l'ordinateur local"
    type = "string"
    many="false"
    required = "false" />
<named
    name = "e"
    helpstring = "Commutateur indiquant le type de l'entrée :
    R  : clef de la branche HKLM de la base de registres (software\\...)
        (valeur par défaut)
    T  : Recherche de toutes les clefs 'DigitalProductID'
    P  : ProductKey
    HL : chaîne hexadécimale de ProductKey (RAW key)
        en notation Little Endian (octets de poids faibles en lers)
    HB : chaîne hexadécimale de ProductKey (RAW key)
        en notation Big Endian (octets de poids forts en lers)"
    type = "string"
```

```
    required = "false" />
<unnamed name = "v"
  helpstring = "Suivant la valeur du commutateur précédent
  - clef de la branche HKLM
    NB: ne pas indiquer ni le préfixe (HKLM)
          ni le nom de l'entrée (DigitalProductID)
        (à encadrer par des guillemets si le nom contient des espaces)
    Si ce paramètre est absent, on retient la clef :
    \SOFTWARE\Microsoft\Windows NT\CurrentVersion
  - ProductKey sous la forme xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
  - chaine hexadécimale (RAW Key)"
  type = "string"
  many="false"
  required = "false" />
<example>
Exemples :
-----
viewpk
  affiche la ProductKey de Windows

viewpk /e:T /c:GRANDBASSAM
  affiche les ProductKey de tous les produits Microsoft
  installés sur l'ordinateur "GRANDBASSAM"

viewpk /c:SASSANDRA /e:R
"SOFTWARE\Microsoft\Office\11.0\Registration\{9017040C-6000-11D3-8CFE-0
150048383C9}"
  affiche la ProductKey de FrontPage 2003 sur l'ordinateur "SASSANDRA"

viewpk /e:HL 8063EFA7DE99C873A76B9F97FCD303
  affiche la ProductKey à partir de cette chaine Little Endian

viewpk /e:P FCKGW-RHQ2-YXRKT-8TG6W-2B7Q8
  affiche la RAWKey à partir de cette ProductKey

Dans tous les cas, il est affiché :
  - la RAWKey en notation Little Endian
  - la RAWKey en notation Big Endian
  - les 2 nombres centraux du productID
    (algorithme valable seulement pour Windows)
-----
</example>
</runtime>
<script language="VBScript">

const HKEY_LOCAL_MACHINE    = &H80000002
Charset24 =
array("B","C","D","F","G","H","J","K","M","P","Q","R","T","V","W","X","
Y","2","3","4","6","7","8","9")
```

```

'           0  1  2  3  4  5  6  7  8  9  10  11  12
13  14  15  16  17  18  19  20  21  22  23
'           0  1  2  3  4  5  6  7  8  9  A  B  C
D   E   F  10  11  12  13  14  15  16  17
Prefix="HKLM\"
KeyDef="software\Microsoft\Windows NT\CurrentVersion"
Entry1="DigitalProductId"
Entry2="ProductID"
Entry3="ProductName"
SW_SHOWNORMAL=1
Dim shell, fso, net, args, Buffer, PK(25),Result(15), Key
Set shell = WScript.CreateObject("WScript.Shell")
Set net    = Wscript.CreateObject("WScript.Network")
Set args   = Wscript.Arguments
Set fso    = WScript.CreateObject("Scripting.FileSystemObject")
nbargs=args.count
TestHost false

Set Named   = WScript.Arguments.Named
Set UnNamed = WScript.Arguments.UnNamed
nu=UnNamed.count
If named.Exists("c") Then Computer=lcase(named.Item("c")) else
Computer=lcase(net.ComputerName)
Set oReg=GetObject("winmgmts:{impersonationLevel=impersonate}!\" &
Computer & "\root\default:StdRegProv")
On Error Resume Next
job="r"
If named.Exists("e") Then job=lcase(named.Item("e"))
Select Case job
'Analyse clef BDR
    Case "r"
        BDR=true
        If nu=0 Then Key=KeyDef else Key=UnNamed(0)
        coderet=oReg.GetBinaryValue(HKEY_LOCAL_MACHINE, Key, Entry1,
Buffer)
        If CodeRet<>0 Then
            Wscript.echo "Erreur BDR"
            Wscript.quit
            End If
        For index = 0 to 14
            PK(index)=Buffer(index+&H34)
        Next
        PrintResult
'Enumération de toutes les clefs
    Case "t"
        On error goto 0
        BDR=true
        rep=shell.Popup("Veuillez patienter SVP." & VBCRLF & "La
recherche des clefs peut prendre un certain temps", _
3, "Clefs de produits Microsoft sur " & Computer,65)
        If rep=2 Then Wscript.quit

```

```
        ExploreKey("Software\Microsoft")
'Analyse de RAWKey ou ProductKey
    Case "hl","hb","p"
        BDR=false
        If nu=0 Then
            WScript.Arguments.ShowUsage
            WScript.Quit
        end if
        Key=UnNamed(0)
        Select Case job
'Little Endian
            Case "hl"
                i1=0
                i2=14
                s=1
'Big Endian
            Case "hb"
                i1=14
                i2=0
                s=-1
            Case "p"
            End Select
        If job<>"p" Then
'RAWKey
            j=0
            For index = i1 To i2 Step s
                Sbyte="&H" & Mid(Key,index*2+1,2)
                PK(j)=CInt(Sbyte)
                j=j+1
            Next
            PrintResult
        Else
'ProductKey
            j=0
            For index = 1 To len(Key)
'        conversion des caractères en base24
                v=valueB24(Mid(Key,index,1))
                If v>=0 Then
                    PK(j)=v
                    j=j+1
                End If
            Next
            If j<>25 Then
                WScript.Arguments.ShowUsage
                WScript.Quit
            End If
'        conversion en binaire
            For i = 0 To 14
                Result(i)=0
            Next
        End If
    End Select
End Sub
```

```
        Next
        NumDigits=24
        ResLen=15
        For i= 0 To 24
            Aux=PK(i)
            m=0
            Do
                Aux=Result(m)*NumDigits + Aux
                Result(m)= Aux AND 255
                m=m+1
                Aux=int(Aux/256)
            Loop Until m >= ResLen
        Next
        For i = 0 To 14
            PK(i)=Result(i)
        Next
        PrintResult
    End If
Case else
    WScript.Arguments.ShowUsage
    WScript.Quit
End Select

Wscript.quit
'-----
'Exploration récursive de la BDR
'recherche des entrées "DigitalProductId"
Sub ExploreKey(CurKey)
    coderet=oReg.GetBinaryValue(HKEY_LOCAL_MACHINE,CurKey,Entry1,Buffer)
    If CodeRet=0 Then
        'Entrée trouvée. Extraction RAWKey
        For index = 0 to 14
            PK(index)=Buffer(index+&H34)
        Next
        Key=CurKey
        PrintResult
        Exit Sub
    end if
Dim arrSubKeys
'Enumération des sous-clefs
coderet=oReg.EnumKey(HKEY_LOCAL_MACHINE,CurKey,arrSubKeys)
If codeRet=0 Then
    If IsArray(arrSubKeys) Then
        For Each subkey In arrSubKeys
            ExploreKey(CurKey & "\" & subkey)
        Next
    End If
end if
End Sub
'-----
Sub PrintResult
```

```
RAWKeyBE=""
RAWKeyLE=""
'Affichage des RAWkeys en Little Endian et en Big Endian
For index = 0 to 14
    RAWKeyBE=hexa(PK(index)) & RAWKeyBE
    RAWKeyLE=RAWKeyLE & hexa(PK(index))
Next
S=""
'Calcul du ProductID (algorithme valable seulement pour Windows)
For index = 0 To 3
    CurByte=PK(Index)
    If index = 3 Then CurByte=CurByte and &H7F
    S=Hexa(CurByte) & S
next
PID=int("&H" &S) /2)
SPID=CStr(PID)
While len(SPID)<9
    SPID="0" & SPID
Wend
SPID1=left(SPID,3)
SPID2=mid(SPID,4)
sigma=0
For i = 1 To len(SPID2)
    sigma=sigma+Asc(mid(SPID2,i,1))-48
Next
m=7-sigma mod 7
SPID=SPID1 & "-" & SPID2 & m
'Conversion RAWKey en ProductKey
S=""
For i = 24 To 0 Step -1
    r = 0
    For j = 14 To 0 Step -1
        r = r * 256 Xor PK(j)
        PK(j) = Int(r/24)
        r = r Mod 24
    Next
    S=CharSet24(r) & S
    If i Mod 5 = 0 And i <> 0 Then S="-" & S
Next
msg=VBCRLF
If BDR Then msg=msg & "HKLM\" & Key & "\" & Entry1
msg=msg & VBCRLF & "ProductKey" & " " & S
msg=msg & VBCRLF & "RAWKey Big Endian" & " " & RAWKeyBE
msg=msg & VBCRLF & "RAWKey Little Endian" & " " & RAWKeyLE
' Calcul du ProductID (central)
' NB : valable seulement pour Windows, l'algorithme semblant
' être différent pour les autres produits
msg=msg & VBCRLF & "ProductID calculé" & " " & "xxxxx-" & SPID & "-
xxxxx"
```

```

If BDR Then
    coderet=oReg.GetStringValue(HKEY_LOCAL_MACHINE, Key, Entry2, SPID)
    If Coderet=0 Then msg=msg & VBCRLF & "ProductID BDR          = " &
SPID
    coderet=oReg.GetStringValue(HKEY_LOCAL_MACHINE, Key, Entry3, SPID)
    If Coderet=0 Then msg=msg & VBCRLF & "ProductName          = " &
SPID
    End If
Wscript.echo msg
End Sub
'-----
'Affichage hexadécimal d'un octet avec 0 non significatif éventuel
Function hexa(n)
ch=hex(n)
If len(ch)<2 then ch="0" & ch
hexa=ch
End Function
'-----
Function valueB24(c)
valueB24=-1
c=ucase(c)
For i = 0 To 23
    If c=CharSet24(i) Then
        valueB24=i
        exit for
    End If
Next
End Function
'-----
'Sous-programme de test du moteur
'Vu les sorties générées, c'est CSCRIPT (et non pas WSCRIPT)
'qui doit être utilisé de préférence
Sub TestHost(force)
dim rep
strappli=lcase(Wscript.ScriptFullName)
strFullName =lcase(WScript.FullName)
i=InStr(1,strFullName,".exe",1)
j=InStrRev(strFullName,"\\",i,1)
strCommand=Mid(strFullName,j+1,i-j-1)
if strCommand<>"cscript" then
    If force then
        Init="Ce script doit être lancé avec CSCRIPT"
    Else
        Init="Il est préférable de lancer ce script avec CSCRIPT"
    End If
    rep=MsgBox(Init & VBCRLF & _
"Cela peut être rendu permanent avec la commande" & VBCRLF & _
"cscript //H:CScript //S /Nologo" & VBCRLF & _
"Voulez-vous que ce soit fait automatiquement?", _
vbYesNo + vbQuestion,strappli)
    if rep=vbYes then

```

```
nomcmd="setscript.bat"
Set ficcmd = fso.CreateTextFile(nomcmd)
ficcmd.writeline "@echo off"
ficcmd.writeline "cscript //H:CScript //S /Nologo"
ficcmd.writeline "pause"
params=""
For i = 0 To nbargs-1
    params=params & " " & args(i)
next
ficcmd.writeline chr(34) & strappli & chr(34) & params
ficcmd.writeline "pause"
ficcmd.close
shell.Run nomcmd, SW_SHOWNORMAL,true
force=true
end if
If force then WScript.Quit
end if
end sub
'-----
</script>
</job>
```

Dans une fenêtre DOS, positionnez vous dans le dossier où le fichier **viewpk.wsf**, puis exécutez la commande :

```
cscript viewpk.wsf
```

Le résultat devrait de la forme suivante :

ProductKey	=	FXXXP-4DFJD-GXXX9-VJXX7-HXXX2
RAWKey Big Endian	=	00D62D7BA2A0036A00BF64300C49D9
RAWKey Little Endian	=	0049FC396000236A83A0027B2DD600
ProductID calculé	=	xxxxx-486-4100400-xxxxx
ProductID BDR	=	00400-OEM-0092002-00006
ProductName	=	Windows 7 Ultimate

+ d'infos

- <http://jc.bellamy.free.fr/fr/systeme.html#clefsproduitsMS>

From:
<https://www.abonnel.fr/> - notes informatique & technologie

Permanent link:
https://www.abonnel.fr/informatique/se/ms-windows/les_clefs_de_produits_microsoft

Last update: 2020/04/17 20:06



