

WshNetwork

Como lo dice su nombre, este objeto nos permitirá realizar determinadas acciones en red:

- Listar unidades de red
- Mapear o remover unidades de red
- Listar impresoras de red
- Conectar y desconectar a impresoras en red
- Obtener datos básicos de red

Como vemos en este pequeño resumen, las funciones de este objeto son todas las que hacen referencia a una red, y se declara por medio de "wscript.network".

Pasemos entonces a ver cada una de sus funciones.

Listar unidades de red

Como vimos anteriormente, sería posible listar las unidades de red, por medio del objeto FSO y su función **drives**, pero el caso es que obtendríamos solamente la letra de la unidad, no así con el objeto network, con el cual además, podemos conseguir la ruta de red de la unidad.

```
Set objnet = createobject("wscript.network")
```

```
Set variable = objnet.enumnetworkdrives
```

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Set netdrive = objnet.enumnetworkdrives
```

```
For each n in netdrive
```

```
    MsgBox n
```

```
next
```

Este ejemplo, no será nada extraño, ya que se utiliza el mismo método que al listar los discos con FSO, es decir, declaramos el objeto, y luego a través de este, obtenemos una colección de unidades, en este caso, unidades de red, para luego por medio de un **for**, recorrer cada uno de los elementos de la colección.

En el caso del objeto network, la colección resulta un tanto especial, y esto es porque esta colección, esta formada por pares de elementos.

Con pares de elementos quiero decir que; si listamos todos los elementos, tendremos primero una unidad (letra), y luego la ruta de red, segundo, otra unidad, y luego, su respectiva ruta de red, etc. Por lo que dicho de otra forma, todos los índices pares del array, son letras, y todos los impares, son rutas de red.

Por lo que, analizando nuestro ejemplo, el msgbox mostraría algo como; X:, \\servidor\usuarios, Y:, \\pcremota\compartido, Z:, \\varios\musica, cada uno de estos, en una ventana de mensaje diferente.

Veamos entonces otro ejemplo, uno donde se vea, como podemos mostrar unas, u otras, es decir, o bien letras, o bien rutas.

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Set netdrive = objnet.enumnetworkdrives
```

```
For i = 0 to netdrive.count -1 step 2
```

```
    MsgBox netdrive(i)
```

```
Next
```

‘en este ejemplo se mostrarán solo las letras

Para empezar, las dos primeras líneas no muestran nada raro, simplemente se declara el objeto, y se obtiene la colección, sí que pasemos al **for** que es donde se ve algo distinto.

Como vemos, iniciamos la variable i en 0 (cero), esto es porque como sabemos, los array comienzan en este valor y además, el primero de ellos debe de ser una letra.

Luego, fijamos el límite por medio de la función **count**, la cual nos muestra cuantos elementos contiene la colección (aplicable a todas las colecciones), y luego a este valor, le restamos **1**, ya que también como hemos dicho antes, los array comienzan por cero, por lo que si la colección contiene 6 elementos, el array irá desde 0 hasta 5.

Y por último, recorremos el **for** de dos en dos (lo cual indicamos con **step 2**) y mostramos un mensaje con la letra.

Al igual que antes, sugiero mirar detenidamente el script para entenderlo mejor, ya que muchas veces, el explicarlo resulta bastante largo para tan pocas líneas de texto.

Mappear unidades de red

Variable.mapnetworkdrive(letra, rutaremota, actualizarperfil, usuario, contraseña)

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Objnet.mapnetworkdrive "G:", \\pcremota\compartido
```

Los dos primeros parámetros son, la letra a la que mapearemos la unidad remota (con el dos puntos incluido), y la ruta remota. Los otros tres parámetros son; true o false para seleccionar si queremos guardar el mapeado solamente para nuestro usuario, y los otros dos, usuario y contraseña en caso de que accedamos a la unidad con los datos un usuario distinto al que ha iniciado sesión (con el que estamos trabajando).

En este ejemplo, solamente he utilizado los dos primeros parámetros ya que los otros tres son opcionales.

Remover unidades de red

Variable.removeNetworkdrive(letra, forzar, actualizarperfil)

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Objnet.removeNetworkdrive "G:", true
```

Este no requiere de mucha explicación, simplemente removemos una unidad de red, para eso, seleccionamos la unidad, true o false para forzar o no, y al igual que antes, si se actualiza o no el perfil de usuario.

Listar impresoras en red

Este método es prácticamente igual al de listar unidades de red, solamente que se cambia la función.

Set variable = objnet.enumPrinterconnections

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Set netprinter = objnet.enumPrinterconnections 'Colección de impresoras
```

```
For each n in netprinter
```

```
    MsgBox n
```

```
next
```

Al igual que para el caso de las unidades de red, este array también viene en pares, donde el primer valor es la localización remota (IP), y la segunda, es el nombre con el que tenemos configurada nuestra impresora (al contrario de cómo pasaba con las unidades de red, donde primero esta la unidad local, y luego la ruta remota)

Nota: Complementos como drivers que agregan una impresora para imprimir en pdf, también son mostrados como remotos.

Las direcciones IP aparecen como IP_x_x_x_x ej: **IP_192_168_15_3**

Conectar a impresora en red

También es muy sencillo y similar a lo visto anteriormente

Variable.addPrinterconnection(nombrelocal, rutaRemota, actualizarperfil, usuario, contraseña)

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Objnet.addPrinterconnection "LPT1", \\servidor\impresoraCompartida
```

by Novlucker

Al igual que antes, simplemente pasamos como parámetros los datos “locales” y “remotos”, y solamente los dos primeros parámetros, ya que los otros son opcionales.

Desconectar impresora en red

Para desconectar una impresora en red, haremos uso de la función **removeprinterconnection**

Variable.removeprinterconnection(nombre, forzar, actualizarperfil)

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Objnet.removeprinterconnection \\servidor\impresoracompartida, True, True
```

Como nombre de impresora, se puede pasar direcciones de red, como lo he hecho en este ejemplo, o bien rutas de impresoras locales, como lo sería LPT1

Información de red

A través de este objeto podemos acceder a algunos datos de red:

- Usuario; Username
- Dominio; Domain
- Nombre de la PC; Computername

Si han leído la parte anterior del tuto, sabrán que se puede acceder a las variables del sistema (comando SET de ms-dos) por medio del objeto wshshell, por lo que también podríamos acceder a estos datos, por lo que este método se presenta como una alternativa más dentro del wsh.

variable.dato

Ejemplo:

```
Set objnet = createobject("wscript.network")
```

```
Msgbox objnet.username
```

```
Msgbox objnet.userdomain
```

```
Msgbox objnet.computername
```

En definitiva y como se ve en el ejemplo, solo hay que utilizar el objeto con su respectiva propiedad.

Nota: Si no se esta en un dominio, el userdomain y computername tendrán el mismo valor.

Así termina la explicación de este escueto pero funcional objeto, y siendo esta la tercera parte, estoy seguro de que no han quedado dudas