Lenguaje de comandos de Automatización de Acceso telefónico
Para archivos de comandos de Acceso telefónico a redes

Contenido

1.0 Información general

Muchos proveedores de servicios Internet y de servicios en línea requieren que introduzca información manualmente para poder establecer la conexión, como por ejemplo su nombre y contraseña. Con los archivos de comandos para Acceso telefónico a redes, puede automatizar este procedimiento.

Un archivo de comandos contiene una serie de comandos, parámetros y expresiones que necesita su proveedor de servicios Internet o de servicios en línea para establecer la conexión y utilizar el servicio. Puede utilizar cualquier editor de texto como el Bloc de notas de Microsoft para crear un archivo de comandos. Una vez creado el archivo de comandos puede asignarlo a una conexión de Acceso telefónico a redes ejecutando la Automatización de Acceso telefónico.

2.0 Estructura básica de un archivo de comandos

Un comando es la instrucción básica que contiene un archivo de comandos. Algunos comandos requieren parámetros que definen lo que el comando debe hacer. Una expresión es una combinación de operadores y argumentos que generan un resultado. Las expresiones se pueden utilizar como valores en cualquier comando. Entre las expresiones están las aritméticas, las comparaciones relacionales y las concatenaciones de cadenas.

El formato básico de un archivo de comandos para Automatización de Acceso telefónico es el siguiente:

```
;
; Un comentario comienza con un punto y coma y continúa
; hasta el final de la siguiente línea.
;

proc main
       ;  Un archivo de comandos puede contener cualquier
       ;  número de variables y comandos

       declaracion de variables

       bloque de comandos

endproc
```

Un archivo de comandos debe tener un procedimiento principal especificado por la palabra clave **proc** y una palabra clave **endproc** coincidente que indica el final del procedimiento.

Debe declarar las variables antes de agregar comandos. Se ejecuta el primer comando en el procedimiento principal y después todos los siguientes comandos se ejecutan en el orden en el que aparezcan en el archivo de comandos. El archivo de comandos finaliza cuando se llega al final de procedimiento principal.

3.0 Variables

Los archivos de comandos pueden contener variables. Los nombres de variables deben comenzar con una letra o un subrayado ('_') y pueden contener cualquier combinación de letras mayúsculas o minúsculas, dígitos y subrayados. No se puede utilizar una palabra reservada como nombre de variable. Para más información, consulte la lista de palabras reservadas incluida al final de este documento.

Debe declarar las variables antes de utilizarlas. Cuando declara una variable también debe definir su tipo. Una variable de cierto tipo sólo puede contener valores de ese mismo tipo. Se pueden utilizar los tres siguientes tipos de variables:

Tipo	Descripción
entero	Un número negativo o positivo, como 7, -12, ó 5698.
cadena	Una serie de caracteres entre comillas; como por ejemplo "Hola a todos" o "Escriba contraseña:"
booleano	Una valor booleano lógico de TRUE o FALSE (verdadero o falso).

Se asignan valores a las variables de acuerdo con la siguiente declaración de asignación:

variable **=** *expresión*

La variable obtiene la expresión analizada.

Ejemplos:

```
integer cuenta = 5
integer paso = (4 * 3)
integer i

boolean Terminado = FALSE

string szIP = (getip 2)

set ipaddr szIP
```

3.1 Variables de sistema

Las variables de sistema se establecen con comandos de automatización o se determinan por la información suministrada al configurar su conexión de Acceso telefónico a redes. Las variables de sistema son de sólo lectura, por lo que no se pueden cambiar en el archivo de comandos. Las variables de sistema son:

Nombre	Tipo	Descripción
$USERID	Cadena	La identificación de usuario para la conexión actual. Esta variable es el valor de nombre de usuario especificado en el cuadro de dialogo 'Conectar con' de Acceso telefónico a redes.

$PASSWORD	Cadena	La contraseña para la conexión actual. Esta variable es el valor de nombre de usuario especificado en el cuadro de dialogo 'Conectar con' de Acceso telefónico a redes.
$SUCCESS	Booleano	Esta variable la establecen ciertos comandos para indicar si el comando se ejecutó satisfactoriamente o no. Un archivo de comandos puede tomar decisiones en función del valor de esta variable.
$FAILURE	Booleano	Esta variable la establecen ciertos comandos para indicar si el comando falló o no. Un archivo de comandos puede tomar decisiones en función del valor de esta variable.

Estas variables se pueden utilizar donde se utilice una expresión de tipo similar. Por ejemplo,

```
transmit $USERID
```

es un comando válido porque $USERID es una variable de tipo cadena.

4.0 Cadenas literales

La Automatización de Acceso telefónico a redes admite secuencias de escape y traducciones de símbolos de intercalación (circunflejo), como se describe a continuación.

Cadena Literal	Descripción
^*char*	Traducción de símbolo de intercalación (circunflejo). Si *char* es una valor entre '@' y '_', la secuencia de caracteres se traduce a un valor de byte único entre 0 y 31. Por ejemplo, ^M se convierte en un retorno de carro. Si *char* es un valor entre a y z, la secuencia de caracteres se convierte a un valor de byte único entre 1 y 26. Si *char* es cualquier otro valor, la secuencia de caracteres no se trata de forma especial.
<cr>	Retorno de carro
<lf>	Avance de línea
\"	Comillas
\^	Símbolo de intercalación simple
\<	'<' simple
****	Barra inversa

Ejemplos:

```
transmit "^M"
transmit "Jorge^M"
transmit "<cr><lf>"
waitfor "<cr><lf>"
```

5.0 Expresiones

Una expresión es una combinación de operadores y argumentos que obtiene un resultado. Las expresiones se pueden utilizar como valores en cualquier comando.

Una expresión puede combinar cualquier variable, entero, cadena o valor booleano con cualesquiera de los operadores unarios y binarios de las siguientes tablas. Todos los operadores unarios tienen la precedencia máxima. La precedencia de los operadores binarios es indicada por su posición en la tabla.

Los operadores unarios son:

Operador	Tipo de operación
-	Menos unario
!	Complemento de uno

Los operadores unarios están en la siguiente tabla en orden de precedencia. Los operadores con mayor precedencia son los primeros:

Operadores	Tipo de operación	Restricciones del tipo
* /	Multiplicativa	Enteros
+ -	Aditiva	Enteros, cadenas (sólo +)
< > <= >=	Relacional	Enteros
== !=	Igualdad	Enteros, cadenas, booleanos
and	Lógica AND	Booleanos
or	Lógica OR	Booleanos

Ejemplos:

```
cuenta = 3 + 5 * 40
transmit "Hola" + " a todos"
delay 24 / (7 - 1)
```

6.0 Comentarios

Todo el texto de una línea que comience con un punto y coma se ignora.

Ejemplos:

```
; esto es un comentario

transmit "hola"           ; transmitir la cadena "hola"
```

7.0 Palabras clave

Las palabras clave especifican la estructura del archivo de comandos. Al contrario que los comandos, no realizan una acción. La lista de palabras clave es la siguiente:

proc *nombre*

Indica el comienzo de un procedimiento. Todos los archivos de comandos deben tener un procedimiento principal (**proc** main). La ejecución del archivo comienza en el procedimiento principal y termina al final del procedimiento principal.

endproc

Indica el final de un procedimiento. Cuando un archivo de comandos se ejecuta hasta la declaración **endproc** del procedimiento principal, Acceso telefónico a redes iniciará PPP o SLIP.

integer *nombre* [**=** *valor*]

Declara una variable de tipo entero. Puede utilizar cualquier expresión o variable numérica para inicializar la variable.

string *nombre* [**=** *valor*]

Declara una variable de tipo cadena. Puede utilizar cualquier variable o cadena literal para inicializar la variable.

boolean *nombre* [**=** *valor*]

Declara una variable de tipo booleano. Puede utilizar cualquier variable o expresión booleana para inicializar la variable.

8.0 Comandos

Todos los comandos son palabras reservadas, lo que significa que no puede declarar variables que tengan el mismo nombre que los comandos. La lista de comandos es la siguiente:

delay *nSegundos*

Hace una pausa del número de segundos especificado por *nSegundos* antes de ejecutar el siguiente comando del archivo.

Ejemplos:

```
delay 2      ; hace una pausa de 2 segundos
delay x * 3  ; hace una pausa de x * 3 segundos
```

getip *valor*

Espera la recepción de una dirección IP desde el equipo remoto. Si su proveedor de servicios Internet envía varias direcciones IP en una cadena, utilce el parámetro *valor* para especificar qué dirección IP debe utilizar el archivo de comandos.

Ejemplos:

```
; obtener la segunda dirección IP
set ipaddr getip 2

; asignar la primera dirección IP recibida a una variable
szAddress = getip
```

goto *etiqueta*

Salta a la posición del archivo de comandos especificada por *etiqueta* y continúa ejecutando los comandos que haya a continuación.

Ejemplo:

```
  waitfor "Prompt>" until 10
  if !$SUCCESS then
      goto BailOut ; salta a BailOut y ejecuta los comandos que
                   ; haya a continuación
  endif

  transmit "bbs^M"
  goto End

BailOut:
  transmit "^M"
```

halt

Detiene el archivo de comandos. Este comando no elimina la ventana de dialogo Terminal. Debe hacer clic en continuar para establecer la conexión. No puede reiniciar el archivo de comandos.

if *condición* **then**
comandos
endif

Ejecuta una serie de *comandos* si *condición es* TRUE (verdadero).

Ejemplo:

```
if $USERID == "Juan" then
       transmit "Juanito^M"
endif
```

etiqueta **:**

Especifica el lugar del archivo de comandos al que saltar. Una etiqueta (label) debe ser un nombre único y respetar las restricciones de los nombres de variables.

set port databits 5 | 6 | 7 | 8

Cambia el número de bits de los bytes que se transmiten y reciben durante la sesión. El número de bits puede estar entre 5 y 8. Si no incluye este comando, Acceso telefónico a redes utilizará la configuración de propiedades especificada para la conexión.

Ejemplo:

```
set port databits 7
```

set port parity none | odd | even | mark | space

Cambia la paridad del puerto durante la sesión. Si no incluye este comando, Acceso telefónico a redes utilizará la configuración de propiedades especificada para la conexión.

Ejemplo:

```
set port parity even
```

set port stopbits 1 | 2

Cambia el número de bits de paro para el puerto durante la sesión. Este número puede ser 1 ó 2. Si no incluye este comando, Acceso telefónico a redes utilizará la configuración de propiedades especificada para la conexión.

Ejemplo:

```
set port stopbits 2
```

set screen keyboard on | off

Activa o desactiva la entrada por teclado en la ventana Terminal de Automatización.

Ejemplo:

```
set screen keyboard on
```

set ipaddr *cadena*

Especifica la dirección IP de la estación de trabajo para la sesión. *Cadena* debe estar en forma de dirección IP.

Ejemplos:

```
szIPAddress = "11.543.23.13"
set ipaddr szIPAddress

set ipaddr "11.543.23.13"

set ipaddr getip
```

transmit *cadena* [**, raw**]

Envía los caracteres especificados por *cadena* al equipo remoto.

El equipo remoto reconocerá las secuencias de escape y las traducciones de símbolo de intercalación (circunflejo) salvo que incluya el parámetro **raw** en el comando. El parámetro **raw** es útil cuando se transmiten variables de sistema $USERID y $PASSWORD que contengan caracteres en el nombre de usuario o en la contraseña, que sin el parámetro **raw** se interpretarían como secuencias de escape o como símbolos de intercalación (circunflejo).

Ejemplos:

```
transmit "slip" + "^M"
transmit $USERID, raw
```

waitfor *cadena* [**, matchcase**] [**then** *etiqueta*
{ **,** *cadena* [**, matchcase**] **then** *etiqueta* }]
[**until** *tiempo*]

Espera hasta que su PC reciba una o más de las cadenas especificadas desde el equipo remoto. El parámetro *cadena* distingue entre mayúsculas y minúsculas salvo que incluya el parámetro **matchcase**.

Si se recibe una cadena coincidente y se utiliza el parámetro *etiqueta*, el comando saltará a la posición del archivo de comandos designada por *etiqueta*.

El parámetro opcional **until** (hasta) *tiempo* define el número máximo de segundos que esperará su PC a recibir la cadena antes de ejecutar el siguiente comando. Sin este parámetro, su PC esperará sin límite.

Si su PC recibe una de las cadenas especificadas, la variable de sistema $SUCCESS se fija como TRUE (verdadero). Por otra parte, se fija como FALSE (falso) si se alcanza el número de segundos especificado por *tiempo* antes de recibir la cadena.

Ejemplos:

```
waitfor "Login:"

waitfor "Password?", matchcase

waitfor "prompt>" until 10

waitfor
       "Login:"     then DoLogin,
       "Password:"  then DoPassword,
       "BBS:"       then DoBBS,
       "Other:"     then DoOther
  until 10
```

while *condición* **do**
comandos
endwhile

Ejecuta la serie de comandos hasta que condición sea FALSE (falso).

Ejemplo:

```
integer cuenta = 0

while cuenta < 4 do
       transmit "^M"
       waitfor "Login:" until 10
       if $SUCCESS then
              goto DoLogin
       endif
       cuenta = cuenta + 1
endwhile
...
```

9.0 Palabras reservadas

Las siguientes palabras están reservadas y no se pueden utilizar como nombres de variables:

and	boolean	databits	delay	
do	endif	endproc		endwhile
even	FALSE	getip	goto	
halt	if	integer	ipaddr	
keyboard	mark	matchcase	none	
odd	off	on	or	
parity	port	proc	raw	
screen	set	space	stopbits	
string	then	transmit	TRUE	
until	waitfor	while		