

# Linguagem de Comandos de Scripts de Acesso Telefónico Para Suporte de Scripts de Acesso Telefónico à Rede

Copyright (c) 1995 Microsoft Corp.

## Índice

1.0	Descrição Geral
2.0	Estrutura Base de um Script
3.0	Variáveis
3.1	Variáveis de Sistema
4.0	Literais de Cadeia
5.0	Expressões
6.0	Comentários
7.0	Palavras-chave
8.0	Comandos
9.0	Palavras Reservadas

## **1.0 Descrição Geral**

Muitos fornecedores de serviços da Internet e serviços online requerem que introduza informações manualmente, como o nome de utilizador e a palavra-passe, para estabelecer uma ligação. Com o suporte de scripts para o Acesso Telefónico à Rede, pode escrever um script para automatizar este processo.

Um script é um ficheiro de texto que contém uma série de comandos, parâmetros e expressões necessárias ao seu fornecedor de serviços da Internet ou serviço online para estabelecer a ligação e utilizar o serviço. Pode utilizar qualquer editor de texto, como o Bloco de Notas da Microsoft, para criar um ficheiro de script. Após ter criado o seu ficheiro de script, pode atribuí-lo a uma ligação de Acesso Telefónico à Rede específica executando a Ferramenta de Scripts de Acesso Telefónico.

## **2.0 Estrutura Base de um Script**

Um comando é a instrução base de um ficheiro de script. Alguns comandos requerem parâmetros que definam melhor o que o comando deve fazer. Uma expressão é uma combinação de operadores e argumentos que criam um resultado. As expressões podem ser utilizadas como valores em qualquer comando. Exemplos de expressões incluem as expressões aritméticas, as comparações relacionais e as concatenações de cadeias.

A forma base de um script para o Acesso Telefónico à Rede é a seguinte:

```
;
; Um comentário começa por um ponto e vírgula e estende-se até ao
; fim da linha.
;

proc main
    ; Um script pode conter qualquer número de variáveis
    ; e comandos

    declarações de variáveis
```

*bloco de comandos*

*endproc*

Um script tem de ter um procedimento principal, especificado pela palavra-chave **proc**, e uma palavra-chave **endproc** correspondente, indicando o fim do procedimento.

Tem de declarar as variáveis antes de adicionar os comandos. O primeiro comando do procedimento principal é executado, e os comandos seguintes são executados pela ordem em que aparecem no script. O script termina quando é atingido o fim do procedimento principal.

### 3.0 Variáveis

Os scripts podem conter variáveis. Os nomes das variáveis têm de começar por uma letra ou um carácter de sublinhado ('\_') e podem conter qualquer sequência de letras maiúsculas ou minúsculas, dígitos e caracteres de sublinhado. Não pode utilizar uma palavra reservada como nome de variável. Para obter mais informações, consulte a lista de palavras reservadas no fim deste documento.

Tem de declarar as variáveis antes de as utilizar. Quando declara uma variável, tem também de definir o respectivo tipo. Uma variável de um determinado tipo apenas pode conter valores desse mesmo tipo. São suportados os três tipos de variáveis seguintes:

<u>Tipo</u>	<u>Descrição</u>
inteiro	Um número negativo ou positivo, como 7, -12 ou 5698.
cadeia	Uma série de caracteres entre aspas; por exemplo, "Olá mundo!" ou "Introduza a palavra-passe:".
booleano	Um valor booleano lógico, ou seja, TRUE (verdadeiro) ou FALSE (falso).

Os valores são atribuídos às variáveis utilizando a seguinte instrução de atribuição:

*variável* = expressão

A variável fica com o valor da expressão calculada.

Exemplos:

```
integer count = 5
integer timeout = (4 * 3)
integer i

boolean bDone = FALSE

string szIP = "getip 2"

set ipaddr szIP
```

### 3.1 Variáveis de Sistema

As variáveis de sistema são definidas por comandos de script ou são determinadas pelas informações que introduz quando configura uma ligação de Acesso Telefónico à Rede. As variáveis de sistema são só de leitura, o que significa que não podem ser alteradas no script. As variáveis de sistema são:

<u>Nome</u>	<u>Tipo</u>	<u>Descrição</u>	
\$USERID	Cadeia	A identificação do utilizador na ligação actual. Esta variável é o valor do nome do utilizador especificado na caixa de diálogo Ligar A do Acesso Telefónico à Rede.	
\$PASSWORD	Cadeia	A palavra-passe da ligação actual. Esta variável é o valor da palavra-passe especificada na caixa de diálogo Ligar A do Acesso Telefónico à Rede.	
\$SUCCESS	Booleano	Esta variável é definida por certos comandos para indicar se o comando foi executado com êxito ou não. Um script pode tomar decisões com base no valor desta variável.	pode
\$FAILURE	Booleano	Esta variável é definida por certos comandos para indicar se o comando falhou ou não. Um script pode tomar decisões com base no valor desta variável.	com

Estas variáveis podem ser utilizadas sempre que for usada uma expressão de tipo semelhante. Por exemplo,

```
transmit $USERID
```

é um comando válido, porque \$USERID é uma variável do tipo cadeia.

## 4.0 Literais de Cadeia

Os scripts para o Acesso Telefónico à Rede suportam sequências de abandono e conversões de circunflexos, tal como é descrito em seguida.

<u>Literal de Cadeia</u>	<u>Descrição</u>
<i>^car</i>	Conversão de circunflexo  Se <i>car</i> for um valor entre '@' e '_', a sequência de caracteres é convertida num valor de byte único entre 0 e 31. Por exemplo, ^M é convertido num retorno de linha (carácter de CR).  Se <i>car</i> for um valor entre a e z, a sequência de caracteres é convertida num valor de byte único entre 1 e 26.  Se <i>car</i> for qualquer outro valor, a sequência de caracteres não é tratada de modo especial.
<cr>	Retorno de linha
<lf>	Avanço de linha
\"	Aspas
\^	Um circunflexo
\<	Um '<'
\\	Barra invertida

Exemplos:

```
transmit "^M"
transmit "Jorge^M"
```

```
transmit "<cr><lf>"
waitfor "<cr><lf>"
```

## 5.0 Expressões

Uma expressão é uma combinação de operadores e argumentos que resulta num valor. As expressões podem ser utilizadas como valores em qualquer comando.

Uma expressão pode combinar qualquer variável ou valores inteiros, de cadeia ou booleanos com qualquer dos operadores unários e binários das tabelas seguintes. Todos os operadores unários têm a prioridade mais alta. A prioridade dos operadores binários é indicada pela respectiva posição na tabela.

Os operadores unários são:

<u>Operador</u>	<u>Tipo de Operação</u>
-	Menos
!	Complemento por um

Os operadores binários estão listados na tabela seguinte pela respectiva ordem de prioridade. Os operadores com prioridade mais elevada estão listados primeiro:

<u>Operadores</u>	<u>Tipo de Operação</u>	<u>Restrições de Tipo</u>
* /	Multiplicação	Inteiros
+ -	Adição	Inteiros, cadeias (apenas +)
< > <= >=	Relacional	Inteiros
== !=	Igualdade	Inteiros, cadeias, booleanos
and	AND lógico	Booleanos
or	OR lógico	Booleanos

Exemplos:

```
count = 3 + 5 * 40
transmit "Olá" + " malta"
delay 24 / (7 - 1)
```

## 6.0 Comentários

Todo o texto de uma linha iniciada por um ponto e vírgula é ignorado.

Exemplos:

```
; isto é um comentário

transmit "olá" ; transmitir a cadeia "olá"
```

## 7.0 Palavras-chave

As palavras-chave especificam a estrutura do script. Ao contrário dos comandos, não executam uma acção. As palavras-chave são listadas seguidamente.

**proc** *nome*

Indica o início de um procedimento. Todos os scripts têm de ter um procedimento principal (**proc** main). A execução do script é iniciada no procedimento principal e termina no final do procedimento principal.

### **endproc**

Indica o fim de um procedimento. Quando o script é executado até à instrução **endproc** do procedimento principal, o Acesso Telefónico à Rede inicia o PPP ou o SLIP.

### **integer** *nome* [ = *valor* ]

Declara uma variável do tipo inteiro. Pode utilizar qualquer expressão ou variável numérica para inicializar a variável.

### **string** *nome* [ = *valor* ]

Declara uma variável do tipo cadeia. Pode utilizar qualquer literal ou variável de cadeia para inicializar a variável.

### **boolean** *nome* [ = *valor* ]

Declara uma variável do tipo booleano. Pode utilizar qualquer expressão ou variável booleana para inicializar a variável.

## **8.0 Comandos**

Todos os comandos são palavras reservadas, o que significa que não pode declarar variáveis com o nome de comandos. Os comandos são listados seguidamente:

### **delay** *nSegundos*

Pára durante o número de segundos especificado por *nSegundos* antes de executar o comando seguinte do script.

Exemplos:

```
delay 2      ; pára durante 2 segundos
delay x * 3  ; pára durante x * 3 segundos
```

### **getip** *valor*

Aguarda a recepção de um endereço de IP do computador remoto. Se o seu fornecedor de serviços da Internet devolver vários endereços de IP numa cadeia, utilize o parâmetro *valor* para especificar o endereço de IP que o script deve usar.

Exemplos:

```
; obter o segundo endereço de IP
set ipaddr getip 2

; atribuir o primeiro endereço de IP recebido a uma variável
szAddress = getip
```

### **goto** *nome*

Passa para a localização do script especificada por *nome* e prossegue com a execução dos comandos que se seguem.

Exemplo:

```
waitfor "Prompt>" until 10
if !$SUCCESS then
    goto BailOut ; salta para BailOut e executa os comandos
                  ; que se seguem
endif

transmit "bbs^M"
goto End

BailOut:
transmit "^M"
```

## halt

Termina o script. Este comando não remove a janela de diálogo do terminal. Tem de fazer clique sobre Continuar para estabelecer a ligação. Não pode reiniciar o script.

**if condição then**  
    *comandos*  
**endif**

Executa a série de *comandos* se (if) a *condição* for VERDADEIRA (TRUE).

Exemplo:

```
if $USERID == "Joao" then
    transmit "Joaozinho^M"
endif
```

*nome* :

Especifica o local no script onde deve continuar a execução de comandos. O nome tem de ser exclusivo e respeitar as convenções de nomenclatura das variáveis.

## set port databits 5 | 6 | 7 | 8

Altera o número de bits nos bytes que são transmitidos e recebidos durante a sessão. O número de bits pode variar entre 5 e 8. Se não incluir este comando, o Acesso Telefónico à Rede utilizará as definições de propriedades especificadas para a ligação.

Exemplo:

```
set port databits 7
```

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

Altera o esquema de paridade da porta durante a sessão. Se não incluir este comando, o Acesso Telefónico à Rede utilizará as definições de propriedades especificadas para a ligação.

Exemplo:

```
set port parity even
```

### **set port stopbits 1 | 2**

Altera o número de bits de paragem da porta durante a sessão. Este número pode ser 1 ou 2. Se não incluir este comando, o Acesso Telefónico à Rede utilizará as definições de propriedades especificadas para a ligação.

Exemplo:

```
set port stopbits 2
```

### **set screen keyboard on | off**

Activa ou desactiva a utilização do teclado na janela de terminal do script.

Exemplo:

```
set screen keyboard on
```

### **set ipaddr cadeia**

Especifica o endereço de IP da estação de trabalho para a sessão. *Cadeia* tem de ter o formato de um endereço de IP.

Exemplos:

```
szIPAddress = "11.543.23.13"
set ipaddr szIPAddress

set ipaddr "11.543.23.13"

set ipaddr getip
```

### **transmit cadeia [ , raw ]**

Envia os caracteres especificados por *cadeia* para o computador remoto.

O computador remoto reconhecerá as sequências de abandono e as conversões de circunflexos, a menos que inclua o parâmetro **raw** no comando. O parâmetro **raw** é útil para transmitir as variáveis de sistema \$USERID e \$PASSWORD quando o nome do utilizador ou a palavra-passe contiver sequências de caracteres que, sem o parâmetro **raw**, seriam interpretadas como sequências de abandono ou circunflexos.

Exemplos:

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

### **waitfor cadeia [ , matchcase ] [ then nome { , cadeia [ , matchcase ] then nome } ] [ until tempo ]**

Aguarda até que o seu computador receba uma ou mais das cadeias especificadas do computador remoto. O parâmetro *cadeia* não é sensível a maiúsculas e minúsculas, a menos que inclua o parâmetro **matchcase**.

Se for recebida uma cadeia correspondente e for utilizado o parâmetro **then nome**, este comando passará, no ficheiro de script, para o local designado por *nome*.

O parâmetro opcional **until tempo** define o número máximo de segundos que o seu computador aguardará pela recepção da cadeia antes de executar o comando seguinte. Sem este parâmetro, o seu computador aguardará para sempre.

Se o seu computador receber uma das cadeias especificadas, a variável de sistema \$SUCCESS será definida como TRUE. Caso contrário, será definida como FALSE se o número de segundos especificado por *tempo* terminar antes de a cadeia ser recebida.

Exemplos:

```
waitfor "Username:"

waitfor "Password:", matchcase

waitfor "lis10-tp>" until 10

waitfor
    "Username:"      then DoLogin,
    "Password:"     then DoPassword,
    "BBS:"           then DoBBS,
    "Outro:"         then DoOther
until 10
```

**while condição do**  
    *comandos*  
**endwhile**

Executa a série de *comandos* até que (until) a *condição* seja FALSA (FALSE).

Exemplo:

```
integer count = 0

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

## 9.0 Palavras Reservadas

As palavras seguintes são reservadas e não podem ser utilizadas como nomes de variáveis.

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	