

Linguagem de comandos de script dial-up para suporte a scripts do dispositivo de discagem

Copyright (c) 1996 Microsoft Corp.
Parte deste material foi fornecida pela Shiva Corporation.

Utilizando este documento

Este documento foi escrito, a princípio, para o provedor de serviços de Internet (ISP – *Internet Service Provider*), a fim de criar scripts que os usuários finais possam editar para suas próprias conexões.

Como utilizar scripts no Internet Explorer 2.1

Com o IEScript.exe, um usuário final pode utilizar um arquivo de script (.scp) para fazer uma conexão. O IEScript acrescenta definições ao arquivo de conexão (.con), como se observa abaixo. Utilize o IEScript para editar um arquivo de conexão—não o edite manualmente.

Nome da Seção: **Script**

São aceitas as seguintes palavras-chave:

ScriptAtivado

Sim/Não Especifica se o utilitário de script será ativado.

NomeDoArquivoDeScript

Uma seqüência de caracteres que especifica a localização completa do arquivo de script associado ao arquivo de conexão.

TerminalDeScript

Sim/Não Especifica se a janela do terminal de script deve ser carregada durante a execução do script.

GravaçãoDoScript

Sim/Não Especifica se o script executável deve entrar em modo de gravação quando ativado.

Exemplos de entradas de arquivos de conexão para scripts

Exemplo 1:

```
[Script]
ScriptAtivado=Sim
NomeDoArquivoDeScript=C:\Shiva\Meuscript.Scp
TerminalDeScript=Sim
GravaçãoDoScript=Não
```

Exemplo 2:

```
[Script]
ScriptAtivado=Sim
NomeDoArquivoDeScript=C:\Shiva\NovoScip.Scp
TerminalDeScript=Sim
GravaçãoDoScript=Sim
```

Observação

- O Exemplo 2 mostra o que aparece enquanto um script está gravando.

Criando um script no Internet Explorer 2.1

Para criar um arquivo de script para usuários finais, você deve primeiramente gravar um arquivo de script com o Scriptor.exe. Depois de criá-lo, o usuário final poderá editá-lo e inserir seu nome e sua senha.

Gravar e executar o script com a janela TTY

- 1 Execute o Scriptor.exe. Normalmente, ele está localizado no mesmo diretório do programa Internet Explorer.
- 2 No menu **Arquivo**, clique em **Gravar**.
- 3 Localize os arquivos de script (.sco) e então selecione um da lista de arquivos. Ou digite um novo nome de arquivo. Quando você começa a gravar, a janela TTY de script é exibida.
- 4 Abra seu grupo de programas Internet Explorer e, em seguida, escolha **Nova conexão**. Siga as instruções em sua tela para criar uma nova conexão. (Ou escolha o ícone de uma conexão que você já tenha feito.)
- 5 Selecione o ícone da conexão. No menu **Arquivo**, escolha **Propriedades**. Certifique-se de que a caixa de verificação denominada **Exibir janela do terminal após a discagem** esteja selecionada.
- 6 Permita que uma conexão ao servidor Internet seja completada. A janela do ShivaRemote Scriptor exibe um aviso "Iniciado!!!"
- 7 Clique na janela de gravação (atuando no momento como a janela do terminal) e digite os dados necessários para efetuar o login no servidor ISP.
- 8 Depois que for efetuado o login, escolha **Fim da gravação** na janela do Scriptor.
- 9 Na janela do ShivaPPP Scriptor, clique em **OK**.
- 10 Para encerrar as comunicações, clique em **Cancelar**, na caixa de diálogo da conexão.

Observação

Quando estiver utilizando um arquivo de script, o usuário deve certificar-se de que a caixa de verificação denominada **Exibir janela do terminal após a discagem** não esteja selecionada.

Quando você terminar de digitar os dados requeridos no script, clique em **Parar**. A janela fecha-se e a negociação PPP normal tem início.

O script criado pelo programa é o seguinte:

```
; C:\SCRIPTS\IE\LOGIN.SCP
; Criação: 1-7-1996 às 17:07:11
;
;
proc main
string szPassword
transmit "^M"
waitfor "Host Name: ",matchcase until 15
transmit "spr01^M"
waitfor "UIC: ",matchcase until 6
transmit "spr053514^M"
waitfor "Password: ",matchcase until 3
if $PASSWORD then
transmit $PASSWORD
transmit "^M"
goto doneTxPassword
endif
```

```
getinput "Senha: " szPassword
transmit szPassword
transmit "^M"
doneTxPassword:
endproc
```

Editar o arquivo de script baseado no arquivo gravado ou criar um arquivo de script a partir do zero

O Scriptor.exe grava um arquivo de script que contém informações específicas do usuário, incluindo seu nome de usuário e senha. Se você é um provedor de serviços de Internet (ISP), pode desejar apresentar um arquivo de script genérico, que possa ser editado e utilizado por qualquer usuário final.

Utilize o exemplo de script a seguir como um guia para criar um script genérico que não contenha informações específicas do usuário.

```
;-----
; Nome do Arquivo:      LOGIN01.SCP
; Data da Criação:     7-18-1996
; Hora da Criação:     12:00:00
;-----

proc main

; Alguns sistemas exigem que uma tecla seja ativada, para
; iniciar os procedimentos de login
;-----

transmit "^M"

; Envia o nome do host codificado pelo provedor de serviços
;-----

waitfor "Host Name:"
transmit "hostname"
transmit "^M"

; Envia o nome do usuário
; Tirado da caixa Nome do Usuário, do arquivo de conexão
; associado
;-----

waitfor "Username:"
transmit $USERID
transmit "^M"

; Envia a senha
; Tirada da caixa Senha, do arquivo de conexão associado
```

```

;-----
waitfor "Password:"
transmit $PASSWORD
transmit "^M"

endproc

```

Informações avançadas sobre scripts

1.0 Visão geral

Muitos provedores de serviços de Internet e serviços on-line requerem que, para estabelecer uma conexão, o usuário forneça manualmente dados como o nome do usuário e a senha. Com o suporte do Script Dial-Up, o usuário pode gravar um script para automatizar o processo.

Um script é um arquivo texto que contém uma série de comandos, expressões e parâmetros requeridos pelo provedor de serviços de Internet ou serviço on-line para estabelecer a conexão e utilizar o serviço. Você pode utilizar qualquer editor de texto, como o Microsoft Notepad, para criar um arquivo de script. Depois de criá-lo, você poderá então associá-lo a uma conexão Script Dial-Up, executando a Ferramenta de Script Dial-Up.

2.0 Estrutura básica de um script

Um comando é a instrução básica que um arquivo de script contém. Alguns comandos requerem parâmetros que definem o que o comando deverá fazer. Uma expressão é uma combinação de operadores e argumentos que criam um resultado. Expressões podem ser utilizadas como valores em qualquer comando. Exemplos de expressões incluem aritméticas, comparações relacionais e concatenação de seqüências.

A seguir, a forma básica de um script dial-up:

```

;
; Um comentário começa com um ponto-e-vírgula e estende-se até
; o final da linha.
;

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

    declarações de variáveis

    bloco de comandos

endproc

```

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

Você precisa declarar as variáveis antes de acrescentar comandos. O primeiro comando no procedimento principal é executado e, então, quaisquer comandos subseqüentes são executados na ordem em que aparecem no script. O script termina quando é alcançado o fim do procedimento principal.

3.0 Variáveis

Os scripts podem conter variáveis. Os nomes das variáveis devem começar com uma letra ou

um sublinhado () e podem conter qualquer sequência de letras maiúsculas ou minúsculas, números e sublinhados. Você não pode utilizar uma palavra reservada como nome de variável. Para obter maiores informações, veja a lista de palavras reservadas no fim deste documento.

Você precisa declarar as variáveis antes de utilizá-las. Quando você declara uma variável, precisa também definir seu tipo. Uma variável de determinado tipo pode conter apenas valores do mesmo tipo. Os três tipos de variáveis a seguir são aceitos:

Tipo Integer (Inteiro)
Descrição Um número negativo ou positivo, como 7, -12 ou 5698.

Tipo String (Seqüência)
Descrição Uma série de caracteres contidos entre aspas duplas, por exemplo: "Olá, mundo!" ou "Forneça a senha:".

Tipo Boolean (Booleano)
Descrição Um valor booleano lógico TRUE ou FALSE.

As variáveis são associadas a valores utilizando-se a instrução de associação a seguir:

variável = expressão

A variável assume o valor da expressão.

Exemplos:

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

boolean bDone = FALSE

string szIP = (getip 2)

set ipaddr szIP
```

O exemplo acima utiliza comandos de script que não são suportados pelo Shiva.

3.1 Variáveis de sistema

As variáveis de sistema são definidas por comandos de script ou são determinadas pelas informações que você fornece quando faz uma conexão. Elas são do tipo somente leitura, o que significa que não podem ser alteradas dentro de um script. As variáveis de sistema são:

Nome \$USERID
Tipo String
Descrição A identificação do usuário para a conexão atual. Esta variável tem o valor do nome do usuário especificado na caixa de diálogo **Conectar a** do programa de conexão

Nome \$PASSWORD
Tipo String
Descrição A senha para a conexão atual. Esta variável contém o *valor do nome do usuário* especificado na caixa de diálogo **Conectar a** do programa de conexão.

Nome \$SUCCESS

<u>Tipo</u>	Boolean
<u>Descrição</u>	Esta variável é definida por certos comandos, para indicar se o comando foi bem-sucedido ou não. Um script pode tomar decisões baseadas no valor desta variável.

<u>Nome</u>	\$FAILURE
<u>Tipo</u>	Boolean
<u>Descrição</u>	Esta variável é definida por certos comandos, para indicar se o comando falhou ou não. Um script pode tomar decisões baseadas no valor desta variável.

Estas variáveis podem ser utilizadas onde quer que uma expressão de tipo semelhante seja utilizada. Por exemplo,

```
transmit $USERID
```

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

4.0 Literais de seqüência

A criação de scripts para o Script Dial-Up suporta seqüências de escape e marcas de conversão, como descrito a seguir:

<u>Literal de seqüência</u>	<u>Descrição</u>
<u>Literal de seqüência</u>	^char
<u>Descrição</u>	Marca de conversão

Se char for um valor entre '@' e '_', a seqüência de caracteres é traduzida como um valor de bytes simples, entre 0 e 31. Por exemplo, ^M é convertido para retorno de carro.

Se char for um valor entre a e z, a seqüência de caracteres é traduzida como um valor de bytes simples, entre 1 e 26.

Se char for qualquer outro valor, a seqüência de caracteres não é tratada especialmente.

<u>Literal de seqüência</u>	<cr>
<u>Descrição</u>	Retorno de carro

<u>Literal de seqüência</u>	<lf>
<u>Descrição</u>	Salto de linha

<u>Literal de seqüência</u>	\"
<u>Descrição</u>	Aspas duplas

<u>Literal de seqüência</u>	\^
<u>Descrição</u>	Circunflexo

<u>Literal de seqüência</u>	\<
<u>Descrição</u>	Sinal "menor que" (<)

<u>Literal de seqüência</u>	\\
<u>Descrição</u>	Barra invertida

Exemplos:

```
transmit "^M"
```

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

5.0 Expressões

Uma expressão é uma combinação de operadores e argumentos que são avaliados para chegar-se a um resultado. As expressões podem ser utilizadas como valores em qualquer comando.

Uma expressão pode combinar qualquer variável, número inteiro, booleano ou de caractere com qualquer operador unário ou binário contidos nas tabelas abaixo. Os operadores unários possuem a maior precedência. A precedência dos operadores binários é dada por sua posição na tabela.

Os operadores unários são:

<u>Operador</u>	-
<u>Tipo de operação</u>	Menos unário

<u>Operador</u>	!
<u>Tipo de operação</u>	Complemento de um

Os operadores binários estão listados na tabela a seguir, em sua ordem de precedência. Os operadores com maior precedência estão listados primeiro:

<u>Operadores</u>	* /
<u>Tipo de operação</u>	Multiplicação
<u>Restrições de tipo</u>	Números inteiros

<u>Operadores</u>	+ -
<u>Tipo de operação</u>	Números inteiros de adição
<u>Restrições de tipo</u>	Seqüências (+ somente)

<u>Operadores</u>	< > <= >=
<u>Tipo de operação</u>	Relacional
<u>Restrições de tipo</u>	Números inteiros

<u>Operadores</u>	== !=
<u>Tipo de operação</u>	Igualdade
<u>Restrições de tipo</u>	Números inteiros, seqüências, booleanos

<u>Operadores</u>	and
<u>Tipo de operação</u>	E Lógico
<u>Restrições de tipo</u>	Booleanos

<u>Operadores</u>	or
<u>Tipo de operação</u>	OU Lógico
<u>Restrições de tipo</u>	Booleanos

Exemplos:

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

6.0 Comentários

Em uma linha, todo o texto que vier depois de um ponto-e-vírgula é ignorado.

Exemplos:

```
; isto é um comentário  
transmit "olá"          ; transmita a cadeia "olá"
```

7.0 Palavras-chave

As palavras-chave especificam a estrutura do script. Ao contrário dos comandos, elas não executam nenhuma ação. As palavras-chave estão listadas abaixo.

proc *nome*

Indica o início de um procedimento. Todos os scripts devem ter um procedimento principal (**proc main**). A execução do Script começa no procedimento principal e pára no final dele.

endproc

Indica o fim de um procedimento. Quando o script chega à instrução **endproc** do procedimento principal, o Script Dial-Up começa um PPP ou um SLIP.

integer *nome* [= *valor*]

Declara uma variável do tipo Integer. Você pode utilizar qualquer expressão numérica ou variável para inicializá-la.

string *nome* [= *valor*]

Declara uma variável do tipo String. Você pode utilizar qualquer literal de seqüência ou variável para inicializá-la.

boolean *nome* [= *valor*]

Declara uma variável do tipo Boolean. Você pode utilizar qualquer expressão booleana ou variável para inicializá-la.

8.0 Comandos

A implementação inicial da linguagem de scripts para o Internet Explorer 2.1 Dialer será um subconjunto da linguagem de comandos de script dial-up, conforme especificado pelo Windows 95.

Todos os comandos são palavras reservadas, o que significa que você não pode declarar variáveis que tenham o mesmo nome que eles. Os comandos são os seguintes:

delay *nSegundos*

Faz uma pausa, do número de segundos especificado por *nSegundos*, antes de executar o próximo comando no script.

Exemplos:

```
delay 2          ; faz uma pausa de 2 segundos  
delay x * 3      ; faz uma pausa de x * 3 segundos
```

goto *nome*

Salta para o lugar no script especificado pelo nome e continua executando os comandos que o seguem.

Exemplo:

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

transmit "bbs^M"
goto End

BailOut:
transmit "^M"
```

halt

Interrompe o script. Este comando não remove a janela da caixa de diálogo do terminal. Você deve clicar em **Continuar** para restabelecer a conexão. Você não pode reiniciar o script.

```
if condição then
    comandos
endif
```

Executa a série de *comandos*, se a *condição* for TRUE (verdadeira).

Exemplo:

```
if $USERID == "John" then
    transmit "Johnny^M"
endif
```

nome:

Especifica o lugar do script para onde a execução será desviada. Um nome deve ser único e deve seguir as convenções para nomenclatura de variáveis.

transmit seqüência [, **raw**]

Envia para um computador remoto os caracteres especificados pela seqüência.

O computador remoto reconhecerá seqüências de escape e marcas de conversão, a menos que você inclua o parâmetro **raw** no comando. O parâmetro **raw** é útil na transmissão das variáveis de sistema \$USERID e \$PASSWORD, sempre que o nome do usuário ou a senha contenham seqüências de caracteres que, sem este parâmetro, seriam interpretadas como seqüências de conversão ou de escape.

Exemplos:

```
transmit "slip" + "^M"
```

```
transmit $USERID, raw
```

```
waitfor seqüência [ , matchcase ] [ then nome  
    { , seqüência [ , matchcase ] then nome } ]  
    [ until tempo ]
```

Espera até que seu computador receba do computador remoto uma ou mais das seqüências especificadas. O parâmetro seqüência não coincide maiúsculas ou minúsculas, a menos que você inclua o parâmetro **matchcase**.

Se uma seqüência coincidente for recebida e o parâmetro **then** nome for utilizado, o comando saltará para o arquivo de script designado por nome.

O parâmetro opcional **until** tempo define o número máximo de segundos que seu computador deve esperar para receber a seqüência, antes de executar o próximo comando. Sem este parâmetro, seu computador irá esperar indefinidamente.

Se o seu computador receber uma das seqüências especificadas, a variável de sistema \$SUCCESS passa a indicar TRUE (verdadeiro). Caso contrário, ela indicará FALSE (falso) se o número de segundos especificado por *tempo* for atingido antes que se receba a seqüência.

Exemplos:

```
waitfor "Login:"  
  
waitfor "Senha?", matchcase  
  
waitfor "prompt>" until 10  
  
waitfor  
    "Login:"      then DoLogin,  
    "Senha:"      then DoPassword,  
    "BBS:"        then DoBBS,  
    "Outros:"     then DoOther  
until 10
```

O comando seguinte é uma extensão Shiva para o conjunto de comandos de script da Microsoft.

```
getinput "Texto de Aviso " szParaReceberEntrada
```

Solicita que o usuário digite dados durante a execução do script.

Exemplo:

```
Getinput "Senha do Sistema" szSenhaDoSistema
```

9.0 Palavras reservadas

As palavras a seguir são reservadas e não podem ser utilizadas como nomes de variáveis:

and	booleandatabits	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	transmitTRUE	
until	waitfor	while	

10.0 Palavras de comando Shiva não-suportadas pelo Scriptor

getip
port databits
port parity
port stopbits
set screen keyboard
ipaddr
while/endwhile