Skriptsprache für DFÜ-Verbindungen
Skriptunterstützung für das Einwählen

Verwendung dieses Dokuments

Dieses Dokument wurde hauptsächlich für den Internet-Dienstanbieter (IDA) geschrieben, der hierdurch Skripts erstellen kann, die Endbenutzer für ihre eigenen Verbindungen verändern können.

So verwenden Sie die Skriptsprache im Internet Explorer

Mit IEscript.exe kann ein Endbenutzer eine Skriptdatei (.scp) verwenden, um eine Verbindung herzustellen. IESCRIPT fügt Einstellungen zur Verbindungsdatei (.con) hinzu, wie weiter unten dargestellt. Verwenden Sie IESCRIPT, um die Verbindungsdatei zu bearbeiten - verändern Sie die Verbindungsdatei nicht manuell.

Abschnitt: **Script**

Die folgenden Schlüsselwörter werden unterstützt:

ScriptEnabled
Yes/No Legt fest, ob das Skriptprogramm aktiviert wird.

ScriptFileName
Eine vollständig qualifizierte Zeichenfolge, die den Speicherungsort der Skriptdatei angibt, die dieser Verbindungsdatei zugeordnet ist.

ScriptTerminal
Yes/No Legt fest, ob das Terminalfenster für Skripts während der Ausführung des Skripts angezeigt werden soll.

ScriptRecord
Yes/No Legt fest, ob die ausführbare Skriptdatei nach ihrem Aufruf in den Aufnahmemodus gehen soll.

Beispiele für Verbindungsdatei-Einträge in Skripts
Beispiel 1:

```
[Script]
ScriptEnabled=Yes
ScriptFileName=C:\Shiva\Myscript.Scp
ScriptTerminal=Yes
ScriptRecord=No
```

Beispiel 2:

```
[Script]
ScriptEnabled=Yes
ScriptFileName=C:\Shiva\NewScrip.Scp
ScriptTerminal=Yes
ScriptRecord=Yes
```

Anmerkung

- Beispiel 2 veranschaulicht die einzelnen Schritte beim Aufzeichnen eines Skripts.

Erstellen eines Skripts in Internet Explorer

Um eine Skriptdatei für Endbenutzer zu erstellen, müssen Sie zunächst einen Skript mit Hilfe des Programms Scripter.exe aufzeichnen. Nachdem Sie diese Datei erstellt haben, können Sie sie bearbeiten und Ihren eigenen Namen und Ihr Kennwort einfügen.

So führen Sie die Aufzeichnung und Wiedergabe eines Skripts im TTY-Fenster durch

1. Laden Sie Scripter.exe. Das Programm ist in der Regel im Verzeichnis von Internet Explorer gespeichert.
2. Klicken Sie im Menü **Datei** auf **Aufzeichnen**.
3. Wählen Sie eine Skriptdatei (.scp) in der Dateiliste, oder geben Sie einen neuen Dateinamen ein. Wenn Sie mit dem Aufzeichnen beginnen, wird das folgende Fenster angezeigt.
4. Öffnen Sie die Programmgruppe für Internet Explorer, und doppelklicken Sie auf **Neue Verbindung**. Befolgen Sie die Anweisungen auf dem Bildschirm, um eine neue Verbindung zu erstellen. (Oder doppelklicken Sie auf das Symbol einer bereits eingerichteten Verbindung.)
5. Klicken Sie auf das Verbindungssymbol. Wählen Sie aus dem Menü **Datei** den Befehl **Eigenschaften**. Stellen Sie sicher, daß das Kontrollkästchen **Terminalfenster nach dem Wählen anzeigen** aktiviert ist.
6. Stellen Sie eine Verbindung zum Internet Server her. Das ShivaRemote-Skriptfenster zeigt die Meldung "Gestartet!!!" an.
7. Klicken Sie auf das Aufzeichnungsfenster (welches jetzt die Funktion eines Terminalfensters hat), und geben Sie dann die Informationen ein, die für die Anmeldung beim IDA-Server benötigt werden.
8. Wenn die Anmeldung beendet ist, klicken Sie im Skriptfenster auf **Aufzeichnung beenden**.
9. Klicken Sie im ShivaPPP-Skriptfenster auf **OK**.
10. Um die Verbindung zu unterbrechen, klicken Sie im Dialogfeld **Verbindung** auf **Abbrechen**.

Anmerkung

Wenn der Benutzer diese Skriptdatei verwendet, sollte das Kontrollkästchen **Terminalfenster nach dem Wählen anzeigen** deaktiviert sein.

Wenn Sie mit der Eingabe der benötigten Informationen im Skript fertig sind, klicken Sie auf **Halt**. Das Fenster wird geschlossen, und der normale PPP-Ablauf beginnt.

Das durch das Programm erstellte Skript sieht folgendermaßen aus:

```
; C:\SCRIPTS\IE\LOGIN.SCP
; Created:  7-1-1996 at 17:07:11
;
;
proc main
string szPassword
transmit "^M"
waitfor "Host Name:  ",matchcase until 15
transmit "spry01^M"
waitfor "UIC: ",matchcase until 6
transmit "spry053514^M"
waitfor "Password: ",matchcase until 3
```

```
if $PASSWORD then
 transmit $PASSWORD
 transmit "^M"
 goto doneTxPassword
endif
getinput "Password: " szPassword
transmit szPassword
transmit "^M"
doneTxPassword:
endproc
```

Bearbeiten einer aufgezeichneten oder Erstellen einer neuen Skriptdatei

Scripter.exe zeichnet eine Skriptdatei mit benutzerspezifischen Informationen, wie z. B. Benutzername und Kennwort, auf. Wenn Sie ein Internet-Dienstanbieter sind, können Sie eine generische Skriptdatei zur Verfügung stellen, die vom Endbenutzer verändert und verwendet werden kann.

Das folgende Beispielskript dient als Richtlinie für die Erstellung eines generischen Skripts, der keine spezifischen Benutzerinformationen enthält.

```
;------------------------------------------------
; File Name:    LOGIN01.SCP
; Date Created: 10-18-1996
; Time Created: 12:00:00
;------------------------------------------------

proc main

; Manche Systeme erfordern einen Tastendruck, um die
; Anmeldeprozedur zu starten
;-------------------------------------------------------
------

transmit "^M"


; Hostname senden, vom Dienstanbieter fest vorgegeben
;------------------------------------------------

waitfor "Hostname:"
transmit "hostname"
transmit "^M"


; Benutzernamen senden
; Wird dem Feld Name' aus der zugeordneten
; Verbindungsdatei entnommen
;-------------------------------------------------------
-

waitfor "Benutzername:"
transmit $USERID
transmit "^M"
```

```
; Kennwort senden
; Wird dem Feld Password' aus der zugeordneten
; Verbindungsdatei entnommen
;-------------------------------------------------------

waitfor "Kennwort:"
transmit $PASSWORD
transmit "^M"

endproc
```

Erweiterte Informationen zum Erstellen von Skripts

1.0 Überblick

Viele Internet-Dienstanbieter und Online-Dienste verlangen, daß der Benutzer seine persönlichen Informationen wie den Benutzernamen und das Kennwort selbst eingibt, damit eine Verbindung hergestellt werden kann. Mit der Skriptunterstützung für das Einwählen kann der Benutzer ein Skript schreiben, der diesen Ablauf automatisiert.

Ein Skript ist eine Textdatei, die eine Folge von Befehlen, Parametern und Ausdrücken enthält, deren Eingabe vom Internet-Dienstanbieter oder Online-Dienst verlangt wird, um eine Verbindung herstellen und den Dienst nutzen zu können. Sie können jeden beliebigen Texteditor, wie zum Beispiel den Editor von Microsoft, verwenden, um eine Skriptdatei zu erstellen. Nachdem Sie Ihre Skriptdatei erstellt haben, können Sie diese einer bestimmten Skript-Einwählverbindung zuordnen, indem Sie das Programm für Einwählskripte laden.

2.0 Grundstruktur eines Skripts

Ein Befehl ist die grundlegende Anweisung in einem Skript. Einige Befehle erfordern Parameter, die genauer definieren, was der Befehl ausführen soll. Ein Ausdruck ist eine Kombination von Operatoren und Argumenten, welche ein Ergebnis erzeugen. Ausdrücke können in jedem Befehl als Werte verwendet werden. Beispiele für Ausdrücke sind arithmetische oder relationale Vergleiche und die Verkettung von Zeichenfolgen.

Die Grundform eines Einwählskripts sieht folgendermaßen aus:

```
;
; Ein Kommentar beginnt mit einem Semikolon
; und reicht bis zum Ende einer Zeile.

proc main
       ; Ein Skript kann eine beliebige
       ; Anzahl von Variablen und Befehlen
       ; enthalten

     Variablendeklarationen

     Befehlsblock

endproc
```

Ein Skript muß eine Hauptprozedur (**Main**) haben, die durch das Schlüsselwort **proc** deklariert und durch das Schlüsselwort **endproc** abgeschlossen wird.

Variablen müssen vor dem Befehlsblock deklariert werden. Der erste Befehl in der Prozedur **Main** lautet **run**. Alle weiteren Befehle werden in der Reihenfolge ihres Auftretens im Skript ausgeführt. Das Skript endet, wenn das Ende der Prozedur **Main** erreicht ist.

3.0 Variablen

Skripte können Variablen enthalten. Variablennamen müssen mit einem Buchstaben oder einem Unterstrich (_) beginnen und können eine beliebige Folge von Groß- oder Kleinbuchstaben, Zahlen und Unterstrichen enthalten. Sie können kein reserviertes Wort als Variablennamen verwenden. Weitere Informationen finden Sie in der Liste der reservierten Wörter am Ende dieses Dokuments.

Variablen müssen vor ihrer Verwendung deklariert werden. Wenn Sie eine Variable deklarieren, müssen Sie auch ihren Typ definieren. Die Variable eines bestimmten Typs kann nur Werte desselben Typs enthalten. Die folgenden drei Variablentypen werden unterstützt:

Typ	Integer
Beschreibung	Eine negative oder positive Zahl, wie z. B. 7, -12, oder 5698.

Typ	String
Beschreibung	Eine Zeichenfolge, die durch doppelte Anführungszeichen eingeschlossen wird, zum Beispiel "Hallo Welt!" oder "Kennwort eingeben:".

Typ	Boolean
Beschreibung	Der logische Wert TRUE oder FALSE.

Den Variablen werden Werte durch folgende Anweisung zugewiesen:

Variable **=** *Ausdruck*

Die Variable erhält den Wert des aufgelösten Ausdrucks.

Beispiele:

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

boolean bDone = FALSE

string szIP = (getip 2)

set ipaddr szIP
```

Das obige Beispiel verwendet Skriptbefehle, die nicht von Shiva unterstützt werden.

3.1 Systemvariablen

Systemvariablen werden von Skriptbefehlen gesetzt oder durch die von Ihnen eingegebenen Informationen festgelegt, wenn Sie eine Verbindung herstellen. Systemvariablen sind schreibgeschützt, das heißt, sie können nicht innerhalb des Skripts verändert werden. Die Systemvariablen sind:

Name	$USERID
Typ	String
Beschreibung	Die Benutzer-ID für die aktuelle Verbindung. Diese Variable enthält als Wert den Benutzernamen, der im Dialogfeld **Verbinden mit** eingegeben wird.

Name	$PASSWORD
Typ	String
Beschreibung	Das Kennwort für die aktuelle Verbindung. Diese Variable enthält als Wert das Kennwort, das im Dialogfeld **Verbinden mit** eingegeben wird.

Name $SUCCESS
Typ Boolean
Beschreibung Diese Variable wird von bestimmten Befehlen gesetzt, um anzugeben, ob der Befehl erfolgreich ausgeführt wurde oder nicht. Ein Skript kann aufgrund dieses Wertes Entscheidungen treffen.

Name $FAILURE
Typ Boolean
Beschreibung Diese Variable wird von bestimmten Befehlen gesetzt, um anzugeben, ob der Befehl fehlgeschlagen ist oder nicht. Ein Skript kann aufgrund dieses Wertes Entscheidungen treffen.

Diese Variablen können überall da verwendet werden, wo ein Ausdruck mit ähnlichem Typ verwendet wird. Zum Beispiel ist

```
transmit $USERID
```

ein gültiger Befehl, da $USERID eine Variable vom Typ **String** ist.

4.0 String-Literale

Die Skriptsprache für Einwählskripte unterstützt Escape-Folgen und Karet-Umsetzungen wie im folgenden beschrieben.

String-Literal **^***Zeichen*
Beschreibung Karet-Umsetzung

Wenn *Zeichen* einen Wert zwischen '@' und '_' darstellt, wird die Zeichenfolge in ein Byte mit einem Wert zwischen 0 und 31 umgesetzt. ^M wird zum Beispiel in ein Zeichen für die Zeilenschaltung umgewandelt.

Wenn *Zeichen* einen Wert zwischen 'a' und 'z' darstellt, wird die Zeichenfolge in ein Byte mit einem Wert zwischen 97 und 122 umgesetzt.

Wenn *Zeichen* einen beliebigen anderen Wert hat, wird die Zeichenfolge nicht gesondert behandelt.

String-Literal **<cr>**
Beschreibung Zeilenschaltung

String-Literal **<lf>**
Beschreibung Neue Zeile

String-Literal **\"**
Beschreibung Doppeltes Anführungszeichen

String-Literal **\^**
Beschreibung Einfaches Karet-Zeichen

String-Literal **\<**
Beschreibung Einfaches kleiner als' (<)

String-Literal ****
Beschreibung Backslash

Beispiele:

```
transmit "^M"
```

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

5.0 Ausdrücke

Ein Ausdruck ist eine Kombination von Operatoren und Argumenten, die zu einem Ergebnis aufgelöst werden. Ausdrücke können als Werte in beliebigen Befehlen verwendet werden.

In einem Ausdruck kann eine Variable oder einen Wert vom Typ Integer, String oder Boolean mit einem der unären oder binären Operatoren aus den folgenden Tabellen kombiniert werden. Alle unären Operatoren haben den höchsten Rang. Die Rangfolge, d. h. Auswertungsfolge der binären Operatoren wird durch die Position innerhalb der Tabelle angegeben.

Unäre Operatoren sind:

Operator	-
Operationstyp	Unäres Minus

Operator	!
Operationstyp	Bit-Komplement

Die binären Operatoren werden in der folgenden Tabelle in der Reihenfolge ihrer Auswertung aufgeführt. Operatoren mit einem höheren Rang werden zuerst genannt:

Operatoren	* /
Operationstyp	Multiplikation
Anwendbar auf Typ	Integer

Operatoren	+ -
Operationstyp	Addition und Subtraktion
Anwendbar auf Typ	Integer, String (nur +)

Operatoren	< > <= >=
Operationstyp	Relational
Anwendbar auf Typ	Integer

Operatoren	== !=
Operationstyp	Gleichheit
Anwendbar auf Typ	Integer, String, Boolean

Operator	and
Operationstyp	Logisches AND
Anwendbar auf Typ	Boolean

Operator	or
Operationstyp	Logisches OR
Anwendbar auf Typ	Boolean

Beispiele:

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

6.0 Kommentare

Jeder Text in der Zeile nach einem Semikolon wird ignoriert.

Beispiele:

```
; Dies ist ein Kommentar
```

```
transmit "hallo"    ; Zeichenfolge "hallo"
             ; übertragen
```

7.0 Schlüsselwörter

Schlüsselwörter geben die Struktur des Skripts an. Anders als Befehle führen sie keine Aktionen durch. Es folgt eine Liste der Schlüsselwörter:

proc *Name*

Bezeichnet den Beginn einer Prozedur. Alle Skripte müssen eine Hauptprozedur (**proc** main) haben. Die Skript-Ausführung beginnt mit der Hauptprozedur und endet mit dem Ende dieser Prozedur.

endproc

Bezeichnet das Ende einer Prozedur. Wenn das Skript bis zur Anweisung **endproc** der Hauptprozedur ausgeführt worden ist, wird automatisch PPP oder SLIP gestartet.

integer *Name* [**=** *Wert*]

Deklariert eine Variable vom Typ Integer. Sie können einen beliebigen numerischen Ausdruck oder eine Variable für ihre Initialisierung verwenden.

string *Name* [**=** *Wert*]

Deklariert eine Variable vom Typ String. Sie können einen beliebigen String-Literal oder eine Variable für ihre Initialisierung verwenden.

boolean *Name* [**=** *Wert*]

Deklariert eine Variable vom Typ Boolean. Sie können einen beliebigen logischen Ausdruck oder eine Variable für ihre Initialisierung verwenden.

8.0 Befehle

Die erste Implementierung der Skriptsprache für das Wählprogramm von Internet Explorer ist eine Teilmenge der in Microsoft Windows(R) 95 definierten Skriptsprache für DFÜ-Verbindungen.

Alle Befehle sind reservierte Wörter, was bedeutet, daß Sie keine Variablen, die Befehle darstellen, mit Namen deklarieren können. Die Befehle lauten folgendermaßen:

delay *nSekunden*

Hält die Ausführung *nSekunden* an, bevor der nächste Befehl im Skript ausgeführt wird.

Beispiele:

```
delay 2       ; hält 2 Sekunden an
delay x * 3   ; hält x * 3 Sekunden an
```

goto *Marke*

Springt zu der Stelle im Skript, die durch *Marke* gekennzeichnet ist, und setzt die Ausführung mit den auf sie folgenden Befehlen fort.

Beispiele:

```
waitfor "Prompt>" until 10
```

```
 if !$SUCCESS then
      goto BailOut   ; springt zu BailOut und führt die
      nachfolgenden Befehle aus
endif

 transmit "bbs^M"
 goto End

BailOut:
 transmit "^M"
```

halt

Bricht das Skript ab. Dieser Befehl entfernt nicht dasTerminalfenster. Sie müssen auf **Fortsetzen** klicken, um die Verbindung herzustellen. Sie können das Skript nicht erneut ausführen.

if *Bedingung* **then**
Befehle
endif

Führt die Folge von *Befehlen* aus, wenn die *Bedingung* den Wert TRUE hat.

Beispiel:

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

Marke **:**

Kennzeichnet die Stelle im Skript, zu der gesprungen wird. Eine Marke muß aus einem eindeutigen Namen bestehen und die Namenskonventionen für Variablen einhalten.

transmit *String* [**, raw**]

Sendet die durch *String* bezeichnete Zeichenfolge an den entfernten Computer.

Der entfernte Computer berücksichtigt Escape-Sequenzen und Karet-Umsetzungen, sofern Sie nicht den Parameter **raw** angeben. Dieser Parameter ist dann erforderlich, wenn Sie die Systemvariablen $USERID und $PASSWORD übertragen und der Benutzername oder das Kennwort Zeichenfolgen enthält, die ohne den Parameter **raw** als Karet- oder Escape-Sequenzen interpretiert würden.

Beispiele:

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

waitfor *String* [**, matchcase**] [**then** *Marke*
{ **,** *String* [**, matchcase**] **then** *Marke* }]
[**until** *Zeitablauf*]

Wartet solange, bis Ihr Computer eine oder mehrere der angegebenen Zeichenfolgen vom entfernten Computer empfangen hat. Der Parameter *String* beachtet nur die Groß-/Kleinschreibung, wenn Sie den Parameter **matchcase** angegeben haben.

Wenn eine der angegebenen Zeichenfolgen empfangen wurde und der Parameter **then** *Marke* verwendet wird, erfolgt ein Sprung zu der Stelle im Skript, der durch *Marke* bezeichnet worden ist.

Der wahlfreie Parameter **until** *Zeitablauf* legt die maximale Anzahl von Sekunden fest, die Ihr Computer für den Empfang der Zeichenfolge benötigt, bevor der nächste Befehl ausgeführt werden kann. Ohne diesen Parameter wartet Ihr Computer eine unbegrenzte Zeit.

Wenn Ihr Computer eine der angegebenen Zeichenfolgen empfangen hat, wird die Systemvariable $SUCCESS auf den Wert TRUE gesetzt. Ansonsten erhält sie den Wert FALSE, wenn die in *Zeitablauf* angegebene Anzahl Sekunden verstrichen ist, bevor die Zeichenfolge empfangen wurde.

Beispiele:

```
waitfor "Anmelden:"

waitfor "Kennwort?", matchcase

waitfor "prompt>" until 10

waitfor
     "Anmelden:"    then DoLogin,
     "Kennwort:"    then DoPassword,
     "BBS:"         then DoBBS,
     "Anderes:"     then DoOther
 until 10
```

Der folgende Befehl ist eine Erweiterung von Shiva zum von Microsoft entworfenen Skript-Befehlssatz.

getinput "*Zeichenfolge* " sz*Eingabe*

Fordert den Benutzer auf, während der Skript-Ausführung Informationen einzugeben.

Beispiel:

```
Getinput "Kennwort eingeben" szSystemPassword
```

9.0 Reservierte Wörter

Die folgenden Wörter sind reserviert und dürfen nicht als Variablennamen verwendet werden:

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	

10.0 Shiva-Befehle, die nicht von der Skriptsprache unterstützt werden

getip

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