

Język skryptów Dial-Up do obsługi skryptów telefonicznych

Copyright (c) 1997 Microsoft Corp.

Niektóre fragmenty zostały dostarczone przez firmę Shiva Corporation.

Korzystanie z tego dokumentu

Ten dokument został napisany przede wszystkim dla dostawców usług internetowych (DUI), aby mogli utworzyć skrypty, które użytkownicy końcowi zmodyfikują do potrzeb własnych połączeń.

Jak używać skryptów w programie Internet Explorer 3.1

Za pomocą programu IEScript.exe użytkownik może stosować pliki skryptów (.scp) do nawiązywania połączeń. IESCRIPPT dodaje ustawienia do pliku połączenia (.con), tak jak pokazano dalej. Programu można również używać do edycji pliku połączenia—nie należy dokonywać edycji w inny sposób.

Nazwa sekcji: **Script**

Dopuszczalne są następujące słowa kluczowe:

ScriptEnabled

Yes/No Określa, czy uaktywnić obsługę skryptów.

ScriptFileName

W pełni określony ciąg podający lokalizację pliku skryptu skojarzonego z tym plikiem połączenia.

ScriptTerminal

Yes/No Określa, czy okno terminalowe skryptu powinno być załadowane podczas odtwarzania skryptu.

ScriptRecord

Yes/No Określa, czy moduł wykonywalny skrypt powinien przejść do trybu rejestracji przy uaktywnianiu.

Przykład wpisów pliku połączenia z obsługą skryptów

Przykład 1:

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

Przykład 2:

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

Uwaga

- Przykład 2 wyświetla, co pojawia się podczas rejestrowania skryptu.

Tworzenie skryptu w programie Internet Explorer 3.1

Aby utworzyć plik skryptu dla użytkowników końcowych, musisz najpierw zarejestrować plik skryptu za pomocą programu Scripter.exe. Po utworzeniu tego pliku, użytkownik końcowy może dokonać jego edycji i wstawić własną nazwę i hasło.

Rejestrowanie i odtwarzanie skryptu w oknie TTY

- 1 Uruchom program Scripter.exe. Zwykle jest on umieszczony w tym samym katalogu, co program Internet Explorer.
- 2 W menu **Plik** kliknij polecenie **Zarejestruj**.
- 3 Zlokalizuj plik skryptu (.scp), a następnie wybierz go z listy plików. Albo wpisz nazwę nowego pliku. Gdy rozpoczynasz rejestrowanie, wyświetlone jest Okno skryptów TTY.
- 4 Otwórz grupę programów Internet Explorer, a następnie wybierz Nowe połączenie. Postępuj zgodnie z instrukcjami pojawiającymi się na ekranie, aby utworzyć nowe połączenie. (Albo wybierz ikonę już utworzonego połączenia).
- 5 Wybierz ikonę połączenia. W menu **Plik** wybierz polecenie **Właściwości**. Upewnij się, że pole wyboru "Wywołaj okno terminala po wybraniu numeru" jest zaznaczone.
- 6 Pozwól na dokończenie połączenia z serwerem internetowym.
W oknie Obsługa skryptów ShivaRemote zostanie wyświetlony monit "Rozpoczęta!!!".
- 7 Kliknij w oknie rejestracji (działającym obecnie jako okno terminalowe) i podaj informacje wymagane do zalogowania się do serwera dostawcy usług internetowych.
- 8 Po zalogowaniu się wybierz w oknie obsługi skryptów przycisk "Zakończ rejestrację".
- 9 W oknie obsługi skryptów ShivaPPP kliknij przycisk "OK".
- 10 Aby zakończyć połączenie, kliknij przycisk "Anuluj" w oknie dialogowym połączenia.

Uwaga

Podczas korzystania z tego pliku skryptu użytkownik powinien upewnić się, że pole wyboru "Wywołaj okno terminala po wybraniu numeru" nie jest zaznaczone.

Gdy zakończysz podawanie wymaganych w skrypcie informacji, kliknij przycisk "Stop". Okno zostanie zamknięte i rozpocznie się zwykle uzgadnianie PPP.

Skrypt utworzony przez program ma następującą postać:

```
; 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 "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 "Password: " szPassword
transmit szPassword
transmit "^M"
doneTxPassword:
endproc
```

Dokonaj edycji pliku skryptu na podstawie zarejestrowanego pliku lub utwórz plik skryptu od początku

Program Scriptor.exe rejestruje plik skryptu, który zawiera konkretne informacje o użytkowniku, w tym jego nazwę i hasło. Jeśli jesteś dostawcą usług internetowych, możesz chcieć podać uniwersalny plik skryptu, który będzie edytowany i używany przez użytkowników końcowych.

Użyj następującego skryptu przykładowego jako wzorca do tworzenia uniwersalnych skryptów, które nie zawierają konkretnych informacji o użytkowniku.

```
;-----  
; Nazwa pliku:      LOGIN01.SCP  
; Data utworzenia: 7-18-1996  
; Godzina utworzenia: 12:00:00  
;-----  
  
proc main  
  
; W niektórych systemach trzeba nacisnąć klawisz, aby zainicjować  
; procedurę logowania  
;-----  
  
transmit "^M"  
  
; Prześlij nazwę hosta zakodowaną przez dostawcę usług  
;-----  
  
waitfor "Host Name:"  
transmit "hostname"  
transmit "^M"  
  
; Prześlij nazwę użytkownika  
; pobraną z pola Nazwa użytkownika ze skojarzonego  
; pliku połączenia  
;-----  
  
waitfor "Username:"  
transmit $USERID  
transmit "^M"  
  
; Prześlij hasło  
; pobrane z pola Hasło ze skojarzonego  
; pliku połączenia  
;-----  
  
waitfor "Password:"  
transmit $PASSWORD  
transmit "^M"  
  
endproc
```

Zaawansowane informacje o obsłudze skryptów

1.0 Przegląd

Wielu dostawców usług internetowych oraz dostawców usług online wymaga, aby użytkownik samodzielnie podał pewne informacje, na przykład nazwę i hasło, aby ustanowić połączenie. Za pomocą obsługi skryptów Dial-Up Networking można zautomatyzować ten proces.

Skrypt jest plikiem tekstowym zawierającym ciąg poleceń, parametry i wyrażenia wymagane przez dostawcę usług internetowych lub dostawców usług online do ustanowienia połączenia i korzystania z usług. Do tworzenia plików skryptów można użyć dowolnego edytora tekstów, na przykład Notatnika Microsoft. Po utworzeniu pliku skryptu można przypisać go do konkretnego połączenia Dial-Up, uruchamiając Narzędzie obsługi skryptów Dial-Up.

2.0 Podstawowa struktura skryptu

Podstawową instrukcją zawartą w pliku skryptu jest polecenie. Niektóre polecenia wymagają parametrów, które dalej definiują działanie polecenia. Wyrażenie jest kombinacją operatorów i argumentów tworzących wynik. Wyrażenia mogą być używane jako wartości w dowolnym poleceniu. Przykładami wyrażen są wyrażenia arytmetyczne, porównania oraz konkatenacje ciągów.

Podstawowa postać skryptu Dial-Up wygląda następująco:

```
;  
; Komentarz rozpoczyna się od średnika i rozciąga się  
; do końca wiersza.  
;  
  
proc main  
    ; W skrypcie może występować dowolna liczba zmiennych  
    ; i poleceń  
  
    deklaracje zmiennych  
  
    blok poleceń  
  
endproc
```

Skrypt musi mieć procedurę główną (main) określoną przez słowo kluczowe **proc** oraz odpowiadające mu słowo kluczowe **endproc** wskazujące koniec procedury.

Przed dodaniem poleceń należy zadeklarować zmienne. Najpierw wykonywane jest pierwsze polecenie głównej procedury, po nim następują polecenia w kolejności, w jakiej umieszczono je w skrypcie. Skrypt kończy się po osiągnięciu końca głównej procedury.

3.0 Zmienne

W skryptach mogą występować zmienne. Nazwy zmiennych muszą rozpoczynać się od litery lub podkreślenia (_) i mogą zawierać dowolną sekwencję małych i wielkich liter, cyfr oraz podkreśleń. Nie można używać słowa zastrzeżonego jako nazwy zmiennej. Więcej informacji można znaleźć na końcu tego dokumentu, gdzie została umieszczona lista słów zastrzeżonych.

Zmienne należy zadeklarować przed ich użyciem. Gdy deklarujesz zmienną, musisz także zdefiniować jej typ. Zmienna określonego typu może zawierać jedynie wartości tego samego typu. Dopuszczalne są następujące trzy typy zmiennych:

<u>Typ</u>	Integer
<u>Opis</u>	Liczba całkowita, ujemna lub dodatnia, na przykład 7, -12 lub 5698.

<u>Typ</u>	String
<u>Opis</u>	Ciąg znaków umieszczonych w podwójnym cudzysłowie — na przykład "Cześć!" lub "Podaj hasło:".

<u>Typ</u>	Boolean
<u>Opis</u>	Wartość logiczna TRUE (prawda) lub FALSE (fałsz).

Zmiennym można przypisywać wartości, używając następującej instrukcji przypisania:

```
zmienna = wyrażenie
```

Zmienna uzyskuje wartość wyrażenia.

Przykłady:

```
integer zliczenia = 5  
integer limitczasu = (4 * 3)  
integer i
```

```
boolean bGotowe = FALSE
```

```
string szIP = (getip 2)
```

```
set ipaddr szIP
```

W powyższym przykładzie stosowane są polecenia skryptów nie obsługiwane przez metodę Shiva.

3.1 Zmienne systemowe

Zmienne systemowe są ustawiane przez polecenia skryptu lub określone przez informacje podane przy konfigurowaniu połączenia. Zmienne systemowe są tylko-do-odczytu, co oznacza, że nie można ich zmieniać w skrypcie. Przykłady zmiennych systemowych to:

<u>Nazwa</u>	\$USERID
<u>Typ</u>	String
<u>Opis</u>	Identyfikacja użytkownika bieżącego połączenia. Zmienna ta zawiera wartość nazwy użytkownika określoną w oknie dialogowym Połącz z połączenia.

<u>Nazwa</u>	\$PASSWORD_
<u>Typ</u>	String
<u>Opis</u>	Hasło bieżącego połączenia. Zmienna ta zawiera wartość hasła określonego w oknie dialogowym Połącz z połączenia.

<u>Nazwa</u>	\$SUCCESS
<u>Typ</u>	Boolean
<u>Opis</u>	Zmienna ta jest ustawiana przez niektóre polecenia, aby wskazać czy wykonanie polecenia udało się czy nie. Na podstawie wartości tej zmiennej skrypt może podejmować decyzje.

<u>Nazwa</u>	\$FAILURE
<u>Typ</u>	Boolean
<u>Opis</u>	Zmienna ta jest ustawiana przez niektóre polecenia, aby wskazać czy wykonanie polecenia zawiodło czy nie. Na podstawie wartości tej zmiennej skrypt może podejmować decyzje.

Zmienne te mogą być używane we wszystkich miejscach, w których używane są wyrażenia podobnego typu. Na przykład

```
transmit $USERID
```

jest prawidłowym poleceniem, ponieważ zmienna \$USERID jest typu String.

4.0 Ciągi dosłowne

Obsługa skryptów Dial-Up dopuszcza stosowanie sekwencji ESC i umożliwia tłumaczenie znaku ^, tak jak opisano dalej:

<u>Ciąg dosłowny</u>	<u>Opis</u>
<u>Ciąg dosłowny</u>	<u>^znak</u>
<u>Opis</u>	Tłumaczenie znaku ^

Jeśli *znak* jest wartością z zakresu między '@' i '_', sekwencja znaków jest tłumaczona na wartość jednobajtową o wartości między 0 i 31. Na przykład, ciąg ^M jest konwertowany na znak powrotu karetki.

Jeśli *znak* jest wartością z zakresu między a i z, sekwencja znaków jest tłumaczona na wartość jednobajtową o wartości między 1 i 26.

Jeśli *znak* ma inną wartość, sekwencja znaków nie jest traktowana specjalnie.

<u>Ciąg dosłowny</u>	<u><cr></u>
<u>Opis</u>	Powrót karetki

<u>Ciąg dosłowny</u>	<u><lf></u>
<u>Opis</u>	Wysuw wiersza

<u>Ciąg dosłowny</u>	<u>\"</u>
<u>Opis</u>	Podwójny cudzysłów

<u>Ciąg dosłowny</u>	<u>\^</u>
<u>Opis</u>	Pojedynczy znak ^

<u>Ciąg dosłowny</u>	<u>\<</u>
<u>Opis</u>	Pojedynczy znak mniejszości (<)

<u>Ciąg dosłowny</u>	<u>\\</u>
<u>Opis</u>	Odwrotna kreska ukośna

Przykłady:

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

5.0 Wyrażenia

Wyrażenie jest kombinacją operatorów i argumentów przyjmującą wartość wyniku. Wyrażenia można stosować w poleceniach, tak jak inne wartości.

Wyrażenie może łączyć w sobie zmienne, wartości typu integer, string lub boolean za pomocą operatorów jedno- i dwuargumentowych przedstawionych w następujących tabelach. Wszystkie operatory jednoargumentowe mają pierwszeństwo. Kolejność wykonywania operatorów dwuargumentowych wskazuje ich kolejność w tabeli.

Operatorami jednoargumentowymi są:

<u>Operator</u>	-
-----------------	---

<u>Typ operacji</u>	Zamiana znaku
<u>Operator</u>	!
<u>Typ operacji</u>	Uzupełnienie do jedynki

Operatory dwuargumentowe są wymienione w następującej tabeli w kolejności wykonywania. Operatory o najwyższym priorytecie są wymienione na początku:

<u>Operatory</u>	* /
<u>Typ operacji</u>	Mnożenie, dzielenie
<u>Ograniczenie typu</u>	Integer
<u>Operatory</u>	+ -
<u>Typ operacji</u>	Dodawanie, odejmowanie
<u>Ograniczenie typu</u>	String (tylko +)
<u>Operatory</u>	< > <= >=
<u>Typ operacji</u>	Porównanie
<u>Ograniczenie typu</u>	Integer
<u>Operatory</u>	== !=
<u>Typ operacji</u>	Równość, nierówność
<u>Ograniczenie typu</u>	Integer, string, boolean
<u>Operatory</u>	and
<u>Typ operacji</u>	Logiczna koniunkcja
<u>Ograniczenie typu</u>	Boolean
<u>Operatory</u>	or
<u>Typ operacji</u>	Logiczna alternatywa
<u>Ograniczenie typu</u>	Boolean

Przykłady:

```
zliczenie = 3 + 5 * 40
transmit "Dzień" + " dobry"
delay 24 / (7 - 1)
```

6.0 Komentarze

Wszystkie teksty w wierszu umieszczone po średniku są ignorowane.

Przykłady:

```
; to jest komentarz

transmit "cześć" ; przesyła ciąg "cześć"
```

7.0 Słowa kluczowe

Słowa kluczowe określają strukturę skryptu. W odróżnieniu od innych poleceń nie wykonują one czynności. Słowa kluczowe wymieniono dalej.

proc *nazwa*

Wskazuje początek procedury. Wszystkie skryptu muszą mieć procedurę główną (**proc** main). Wykonywanie skryptu rozpoczyna się od początku procedury głównej i kończy na jej końcu.

endproc

Wskazuje koniec procedury. Gdy skrypt dociera do instrukcji **endproc** głównej

procedury, rozpoczyna się działanie protokołu PPP lub SLIP.

integer *nazwa* [= *wartość*]

Deklaruje zmienną typu całkowitego. Możesz użyć dowolnego wyrażenie lub zmiennej do zainicjowania zmiennej.

string *nazwa* [= *wartość*]

Deklaruje zmienną typu ciąg. Możesz użyć dowolnego ciągu dosłownego lub zmiennej do zainicjowania zmiennej.

boolean *nazwa* [= *wartość*]

Deklaruje zmienną typu logicznego. Możesz użyć dowolnego wyrażenia logicznego lub zmiennej do zainicjowania zmiennej.

8.0 Polecenia

Wstępna implementacja języka skryptów w programie Internet Explorer 3.1 Dialer będzie podzbiorem Języka poleceń skryptów Dial-Up określonego w systemie Windows 95.

Wszystkie polecenia są słowami zastrzeżonymi, co oznacza, że nie można deklarować zmiennych o takich samych nazwach co polecenia. Polecenia są następujące:

delay *nSekund*

Wstrzymuje skrypt na liczbę sekund określoną przez argument *nSekund* przed przejściem do następnego polecenia w skrypcie.

Przykłady:

```
delay 2 ; wstrzymuje na 2 sekundy
delay x * 3 ; wstrzymuje na x * 3 sekund
```

goto *etykieta*

Przeskakuje do miejsca w skrypcie określonemu przez *etykieta* i kontynuuje wykonywanie polecenia występującego po nim.

Przykład:

```
waitfor "Prompt>" until 10
if !$SUCCESS then
    goto BailOut ; przeskakuje do etykiety BailOut i wykonuje polecenia
                    ; występujące po niej
endif

transmit "bbs^M"
goto End

BailOut:
transmit "^M"
```

halt

Zatrzymuje skrypt. Polecenie to nie usuwa okna terminalowego. Musisz kliknąć przycisk "Dalej", aby ustanowić połączenie. Nie możesz rozpocząć skryptu od początku.


```
if warunek then
    polecenia
endif
```

Wykonuje ciąg *poleczeń*, jeśli *warunek* ma wartość TRUE (prawda).

Przykład:

```
if $USERID == "Jan" then
    transmit "Jan^M"
endif
```

etykieta :

Określa miejsce w skrypcie, do którego ma nastąpić skok. Etykieta musi mieć unikatową nazwę zgodną z konwencjami nadawania nazw zmiennym.

transmit *ciąg* [, **raw**]

Przesyła znaki określone przez *ciąg* do komputera zdalnego.

Komputer zdalny rozpozna sekwencje ESC i tłumaczenia znaku ^, o ile w poleceniu nie jest umieszczony parametr **raw**. Parametr ten jest przydatny do przesyłania zmiennych systemowych \$USERID i \$PASSWORD, gdy nazwa użytkownika lub hasło zawierają sekwencje znaków, które bez parametru **raw** byłyby interpretowane jako sekwencje znaków ^ i ESC.

Przykłady:

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

waitfor *ciąg* [, **matchcase**] [**then** *etykieta* { , *ciąg* [, **matchcase**] **then** *etykieta* }] [**until** *czas*]

Oczekuje aż komputer odbierze z komputera zdalnego jeden lub kilka z podanych ciągów. W parametrze *ciąg* wielkość liter nie jest rozróżniana. o ile nie jest umieszczony parametr **matchcase**.

Jeśli odebrany zostanie odpowiedni ciąg i jest używany parametr **then etykieta**, polecenie to spowoduje przeskok do miejsca w pliku skryptu oznaczonego przez *etykieta*.

Opcjonalny parametr **until czas** definiuje maksymalną liczbę sekund oczekiwania przez komputer na odebranie ciągu przed przejściem do następnego polecenia. Bez tego parametru komputer czeka przez czas nie określony.

Jeśli komputer odbiera jeden z podanych ciągów, zmiennej systemowej \$SUCCESS nadawana jest wartość TRUE. W przeciwnym przypadku nadawana jest wartość FALSE, jeśli liczba sekund określonych przez parametr *czas* upływa przed odebraniem ciągu.

Przykłady:

```
waitfor "Login:"
waitfor "Password?", matchcase
```

```

waitfor "prompt>" until 10

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

```

Następujące polecenie jest rozszerzeniem Shiva zestawu poleceń skryptów Microsoft.

getinput "Wyświetl ciąg" szToReceiveInput

Monituje użytkownika o podanie informacji podczas wykonywania skryptu.

Przykład:

```
Getinput "Hasło systemowe" szSystemPassword
```

9.0 Słowa zastrzeżone

Następujące słowa są zastrzeżone i nie można ich używać jako nazw zmiennych:

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	transmitTRUE	
until	waitfor	while	

10.0 Słowa poleceń Shiva nie obsługiwane przez skrypty

```

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

```