

Kommandospråk för fjärranslutningsskript för Nummersändaren

Copyright (c) 1997 Microsoft Corp.
Vissa delar av detta material har levererats av Shiva Corporation.

Vilka ska läsa detta dokument

Det här dokumentet har skrivits främst för Internet-leverantörer (ISP) för att de ska kunna skapa skriptfiler som slutanvändarna ska kunna redigera för sina anslutningar.

Så här använder du skript i Internet Explorer

Med IEScript.exe kan en slutanvändare använda en skriptfil (.scp) för att upprätta en anslutning. IESCRIPT lägger till inställningar i anslutningsfilen (.con) enligt nedan. Använd IESCRIPT när du redigerar anslutningsfilen. Redigera aldrig anslutningsfilen manuellt.

Avsnittets namn: **Skript**

Du kan använda följande nyckelord:

ScriptEnabled

Yes/No Anger om skriptfunktionen ska aktiveras.

ScriptFileName

En fullständig sökvägssträng som anger placeringen av den skriptfil som kopplats till den här anslutningsfilen.

ScriptTerminal

Yes/No Anger om terminalfönstret för skript ska läsas in under skriptuppspelningen.

ScriptRecord

Yes/No Anger om den körbara filen för skriptet ska ställas i inspelningsläge vid aktivering.

Exempel på anslutningsfilens poster för skriptfunktionen

Exempel 1:

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

Exempel 2:

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

ScriptRecord=Yes

Obs!

- Exempel 2 visar vad du ser när ett skript spelas in.

Skapa ett skript i Internet Explorer

Om du vill skapa en skriptfil för slutanvändare, måste du först spela in en skriptfil med Scripter.exe. När du har skapat den filen kan slutanvändaren redigera den och infoga sitt eget namn och lösenord.

Spela in och spela upp skriptet med TTY-fönstret

- 1 Kör Scripter.exe. Filen finns i allmänhet i samma katalog som programmet Internet Explorer.
- 2 Klicka på Spela in på Arkiv-menyn.
- 3 Ange platsen för skriptfilen och markera sedan skriptfilen (.scp) i fillistan, eller skriv ett nytt filnamn. När du startar inspelningen visas fönstret Skriptfunktionen.
- 4 Öppna programgruppen Internet Explorer och välj Ny anslutning. Följ sedan instruktionerna på skärmen och skapa en ny anslutning (eller välj en anslutningsikon som du redan har skapat).
- 5 Välj anslutningsikonen. Välj Egenskaper på Arkiv-menyn. Se till att kryssrutan Öppna ett terminalfönster efter numret slagits är markerad.
- 6 Låt anslutningen till Internet-servern fullföljas.
I fönstret ShivaRemote Scripter visas prompten Started!!!.
- 7 Klicka i inspelningsfönstret (som nu fungerar som terminalfönster) och ange därefter den information som behövs för inloggning på ISP-servern.
- 8 När inloggningen är klar klickar du på Avsluta inspelning i fönstret Scripter.
- 9 Klicka på OK i fönstret ShivaPPP Scripter.
- 10 När du vill avsluta kommunikationen klickar du på Avbryt i dialogrutan för anslutningen.

Obs!

När den här skriptfilen används måste användaren kontrollera att kryssrutan Öppna ett terminalfönster efter numret slagits inte är markerad.

När du har skrivit den nödvändiga informationen i skriptfilen klickar du på Stanna. Fönstret stängs då och den vanliga PPP-kommunikationen börjar.

Det skript som skapats av programmet ser ut så här:

```
; 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

```

Redigera en skriptfil som skapats genom inspelning eller skapa en helt ny skriptfil

Scripter.exe spelar in en skriptfil som innehåller användarspecifik information, inklusive användarnamn och lösenord. Om du är en Internet-leverantör (ISP) kan du vilja skapa en generell fil som kan redigeras och användas av vilken slutanvändare som helst.

Använd innehållet i följande exempelskript som riktlinjer för hur du skapar en generell skriptfil som inte innehåller användarspecifik information.

```

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

proc main

; Vissa system kräver en tangenttryckning för att
; starta inloggningsproceduren
;-----

transmit "^M"

; Skicka värddatorns namn som är hårdkodat av
; leverantören
;-----

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

; Skicka användarnamn
; som hämtas från rutan Användarnamn i den kopplade
; anslutningsfilen
;-----

```

```

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

; Skicka lösenord
; som hämtas från rutan Lösenord i den kopplade
; anslutningsfilen
;-----

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

endproc

```

Avancerad skriptinformation

1.0 Översikt

Många Internet-leverantörer och online-tjänster kräver att användaren manuellt ska skriva informationen, t ex användarnamn och lösenord, när anslutningen ska upprättas. Med funktioner för Fjärranslutningsskript kan en användare skapa ett skript som automatiserar denna process.

Ett skript är en textfil som innehåller ett antal kommandon, parametrar och uttryck som krävs av Internet-leverantören eller online-tjänsten för att en anslutning ska kunna upprättas och tjänsten användas. Du kan använda vilken textredigerare som helst, t ex Anteckningar, när du skapar en skriptfil. När du har skapat skriptfilen kan du koppla den till en särskild anslutning genom att köra Verktyg för att ringa upp med hjälp av skript.

2.0 Grundstruktur för ett skript

Ett kommando är den grundinstruktion som ett skript innehåller. Vissa kommandon kräver parametrar som ytterligare definierar vad kommandot ska göra. Ett uttryck är en kombination av operatörer och argument som skapar ett resultat. Uttryck kan användas som värden i kommandon. Exempel på uttryck är aritmetiska beräkningar, jämförelser och strängsammansättningar.

Grundformen för ett fjärranslutningsskript är enligt följande:

```

;
; En kommentar inleds med ett semikolon och sträcker sig
; till slutet av raden.
;

proc main
    ; Ett skript får innehålla det antal variabler och
    ; kommandon som behövs

    variabeldeklaration

    block med kommandon

endproc

```

Ett skript måste innehålla en huvudprocedur, som anges av nyckelordet **proc** och som avslutas med ett matchande nyckelord **endproc**.

Alla variabler måste deklarerars före första kommandot. Det första kommandot i huvudproceduren körs och därefter körs de följande kommandona i den ordning de är inlagda i skriptet. Skriptet avslutas när huvudprocedures slut nås.

3.0 Variabler

Ett skript kan innehålla variabler. Alla variabelnamn måste inledas med en bokstav eller ett understreck (`_`) och får innehålla versaler, gemener, siffror och understreck. Du kan inte använda ett reserverat ord som variabelnamn . Mer information finns i listan över reserverade ord i slutet av detta dokument.

En variabel måste deklarerars innan den kan användas. När du deklarerar en variabel måste du också ange dess typ. En variabel av en viss typ kan bara innehålla värden som matchar typen. Följande tre typer av variabler kan användas:

<u>Typ</u>	Heltal (Integer)
<u>Beskrivning</u>	Ett negativt eller positivt heltal, t ex 7, -12 eller 5698.
<u>Typ</u>	Sträng (String)
<u>Beskrivning</u>	En följd av tecken som omges av citattecken, t ex "Hej svejs!" eller "Skriv ditt lösenord:".
<u>Typ</u>	Boolesk (Boolean)
<u>Beskrivning</u>	Ett av de logiska booleska värdena TRUE eller FALSE.

Variabler tilldelas värden med följande tilldelningssats:

```
variabel = uttryck
```

Variabeln får resultatet av uttryckets värde.

Exempel:

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

boolean bDone = FALSE

string szIP = (getip 2)

set ipaddr szIP
```

I ovanstående exempel används skriptkommandon som inte stöds av Shiva.

3.1 Systemvariabler

Systemvariablers värden anges av skriptkommandon eller av den information som du skriver när du upprättar en anslutning. Systemvariabler är skrivskyddade, vilket innebär att de inte kan ändras inom skriptet. Systemvariablerna är följande:

<u>Namn</u>	\$USERID
<u>Typ</u>	Sträng
<u>Beskrivning</u>	Användarens identifikation för den aktuella anslutningen. Den här variabelns värde är det användarnamn som angetts i dialogrutan Anslut till i anslutningsprogrammet.
<u>Namn</u>	\$PASSWORD_
<u>Typ</u>	Sträng
<u>Beskrivning</u>	Lösenordet för den aktuella anslutningen. Den här variabelns värde är det lösenord som angetts i dialogrutan Anslut till i anslutningsprogrammet.
<u>Namn</u>	\$SUCCESS
<u>Typ</u>	Boolesk
<u>Beskrivning</u>	Den här variabeln tilldelas ett värde av vissa kommandon för att ange om ett kommando har kunnat utföras. Ett skript kan välja olika alternativ beroende på värdet av den här variabeln.
<u>Namn</u>	\$FAILURE
<u>Typ</u>	Boolesk
<u>Beskrivning</u>	Den här variabeln tilldelas ett värde av vissa kommandon för att ange om ett kommando inte har kunnat utföras. Ett skript kan välja olika alternativ beroende på värdet av den här variabeln.

De här variablerna kan alltid användas när ett uttryck av liknande typ används. Till exempel är

```
transmit $USERID
```

ett giltigt kommando eftersom \$USERID är en variabel av strängtyp.

4.0 Strängtecken

Skriptfiler för att ringa upp med skript kan hantera ESC-sekvenser och cirkumflex-konverteringar enligt nedan:

<u>Strängtecken</u>	<u>Beskrivning</u>
<u>Strängtecken</u>	<i>^tecken</i>
<u>Beskrivning</u>	Cirkumflex konvertering

Om *tecken* har ett värde mellan '@' och '_' konverteras teckensekvensen till ett enkelt byte-värde mellan 0 och 31. Till exempel konverteras ^M till en vagnretur.

Om *tecken* har ett värde mellan 'a' och 'z' konverteras teckensekvensen till ett enkelt byte-värde mellan 1 och 26.

Om *tecken* har något annat värde sker ingen bearbetning alls.

<u>Strängtecken</u>	<cr>
<u>Beskrivning</u>	Vagnretur
<u>Strängtecken</u>	<lf>
<u>Beskrivning</u>	Radmatning

<u>Strängtecken</u>	\"
<u>Beskrivning</u>	Citattecken
<u>Strängtecken</u>	\^
<u>Beskrivning</u>	Enkel cirkumflex
<u>Strängtecken</u>	\<
<u>Beskrivning</u>	Enkelt tecken mindre än (<)
<u>Strängtecken</u>	\\
<u>Beskrivning</u>	Omvänt snedstreck (\)

Exempel:

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

5.0 Uttryck

Ett uttryck är en kombination av operatörer och argument som skapar ett resultat vid en utvärdering. Uttryck kan användas som värden i alla kommandon.

I ett uttryck kan du kombinera variabler eller heltal, strängar eller booleska värden med enställiga eller tvåställiga operatörer enligt följande tabell. Alla enställiga operatörer har högsta prioritet. Prioriteten mellan de tvåställiga operatörerna anges av deras placering i tabellen.

De enställiga operatörerna är:

<u>Operator</u>	-
<u>Typ av operation</u>	Enställt minus

<u>Operator</u>	!
<u>Typ av operation</u>	Komplement

De tvåställiga operatörerna räknas upp i följande tabell i sin prioritetsordning. En operatör med högre prioritet kommer före en med lägre:

<u>Operatorer</u>	* /
<u>Typ av operation</u>	Multiplikation/division
<u>Typbegränsningar</u>	Heltal

<u>Operatorer</u>	+ -
<u>Typ av operation</u>	Addition/subtraktion
<u>Typbegränsningar</u>	Strängar (+ endast heltal)

<u>Operatorer</u>	< > <= >=
<u>Typ av operation</u>	Relationer
<u>Typbegränsningar</u>	Heltal

<u>Operatorer</u>	== !=
<u>Typ av operation</u>	Likhet
<u>Typbegränsningar</u>	Heltal, strängar, boolesk typ

<u>Operatorer</u>	and
-------------------	-----

Typ av operation Logiskt AND
Typbegränsningar Boolesk typ

Operatorer or
Typ av operation Logiskt OR
Typbegränsningar Boolesk typ

Exempel:

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

6.0 Kommentarer

All text efter ett semikolon på en rad ignoreras.

Exempel:

```
; Detta är en kommentar
transmit "hej"                ; Skicka strängen "hej"
```

7.0 Nyckelord

Nyckelord anger strukturen i ett skript. Till skillnad från kommandon utför de ingen åtgärd. Följande nyckelord finns:

proc *namn*

Anger början på en procedur. Alla skript måste innehålla en huvudprocedur (**proc main**). Körningen av skriptet startas i början av huvudproceduren och avslutas i slutet av den.

endproc

Anger slutet på en procedur. När skriptet har körts till satsen **endproc** för huvudproceduren startas PPP eller SLIP av Fjärranslutningsskript.

integer *namn* [= värde]

Deklarerar en variabel av typen heltal. Du kan använda ett numeriskt uttryck eller en variabel för att initiera variabeln.

string *namn* [= värde]

Deklarerar en variabel av typen sträng. Du kan använda vilken teckensträng eller variabel som helst för att initiera variabeln.

boolean *namn* [= värde]

Deklarerar en boolesk variabel. Du kan använda ett booleskt uttryck eller en variabel för att initiera variabeln.

8.0 Kommandon

Den första implementeringen av skriptspråket för Nummersändaren i Internet Explorer kommer att utgöra en delmängd av kommandospråket för fjärranslutningsskript i Microsoft Windows(R) 95.

Alla kommandon är reserverade ord, vilket innebär att du inte kan deklarera variabler som har samma namn som ett kommando. Följande kommandon finns:

delay *nSekunder*

Gör en paus under det antal sekunder som anges av *nSekunder* innan nästa kommando i skriptet körs.

Exempel:

```
delay 2      ; Gör en paus i 2 sekunder
delay x * 3  ; Gör en paus i x * 3 sekunder
```

goto *etikett*

Hoppar till den plats i skriptet som anges av *etikett* och fortsätter körningen med kommandona efter *etikett*.

Exempel:

```
waitfor "Prompt>" until 10
if !$SUCCESS then
    goto BailOut ; Hoppar till BailOut och kör de kommandon
                  ; som följer efter etiketten
endif

transmit "bbs^M"
goto End

BailOut:
transmit "^M"
```

halt

Stoppar körningen av skriptet. Kommandot tar inte bort fönstret för terminaldialogen. Du måste klicka på Fortsätt för att upprätta anslutningen. Du kan inte starta om skriptet.

if *villkor* then

kommandon

endif

Utför ett antal *kommandon* om resultatet av *villkor* är TRUE.

Exempel:

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

***etikett*:**

Anger målet för ett hopp i skriptet. En etikett måste vara ett unikt namn som följer

namnreglerna för variabler.

transmit *sträng* [, **raw**]

Skickar de tecken som anges av *sträng* till fjärrdatorn.

Fjärrdatorn känner igen ESC-sekvenser och cirkumflex-konverteringar om du inte tar med parametern **raw** i kommandot. Parametern **raw** är användbar när du t ex överför systemvariablerna \$USERID och \$PASSWORD när användarnamn eller lösenord innehåller teckensekvenser som skulle tolkas som ESC-sekvenser och cirkumflex-konverteringar om du inte angav parametern **raw**.

Exempel:

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

waitfor *sträng* [, **matchcase**] [**then** *etikett* { , *sträng* [, **matchcase**] **then** *etikett* }] [**until** *sekunder*]

Väntar tills datorn mottar en eller flera av de angivna strängarna från fjärrdatorn. Parametern *sträng* är inte skiftlägeskänslig om du inte tar med parametern **matchcase**.

Om en matchande sträng tas emot och parametern **then etikett** har angetts, ger detta kommando ett hopp till den plats i skriptfilen som anges av *etikett*.

Den valfria parametern **until sekunder** anger det antal sekunder som datorn som längst ska vänta på att ta emot strängen innan den kör nästa kommando. Om du inte anger den här parametern kan datorn bli stående i vänteläge hur länge som helst.

Om datorn tar emot någon av de angivna strängarna tilldelas systemvariabeln \$SUCCESS värdet TRUE. I annat fall tilldelas den värdet FALSE, om det antal sekunder har gått, som anges av *sekunder*, utan att strängen har tagits emot.

Exempel:

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

Följande kommando är en Shiva-utvidgning av Microsofts kommandouppsättning för skriptfiler.

getinput "VisaSträng " szToReceiveInput

Uppmanar användaren att skriva information när skriptfilen körs.

Exempel:

```
Getinput "Ange lösenord:" szSystemPassword
```

9.0 Reserverade ord

Följande ord är reserverade och kan inte användas som variabelnamn:

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 Kommandoord i Shiva som inte kan hanteras av Scripter

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