

# Skriptkommandospråk för Fjärranslutning Support för Fjärranslutningsskript

Copyright (c) 1995 Microsoft Corp.

## **Innehållsförteckning**

1.0	Översikt
2.0	Grundläggande struktur
3.0	Variabler
3.1	Systemvariabler
4.0	Strängkonstanter
5.0	Uttryck
6.0	Kommentarer
7.0	Nyckelord
8.0	Kommandon
9.0	Reserverade ord

## **1.0 Översikt**

Flera Internet service providers och online-tjänster kräver att du manuellt skriver in information, som tex användarnamn och lösenord, för att kunna upprätta en kontakt. Med hjälp av Support för Fjärranslutningsskript kan du skriva skript som automatiskt genomför den processen.

Ett skript är en textfil som innehåller ett antal kommandon, parametrar och uttryck som krävs av din Internet service provider eller online-tjänst för att upprätta en kontakt och för att kunna använda tjänsten. För att skapa en skriptfil använder du en textredigerare, som tex Microsoft Notepad. När skriptfilen är skapad kan du tilldela den till en specifik Fjärranslutning genom att köra Skript-verktyg för Fjärranslutning.

## **2.0 Grundläggande struktur**

Kommandon är en grundläggande instruktion i en skriptfil. Vissa kommandon kräver parametrar som definierar vad kommandot ska göra. Uttryck är en kombination av operatorer och argument vars värde kan användas i kommandon. Exempel på uttryck är beräkningar, relationsjämförelser och sammanlänkning av strängar.

Grundläggande kod i ett skript för Fjärranslutning:

```
;
; A comment begins with a semi-colon and extends to
; the end of the line.
;

proc main
    ; A script can have any number of variables
    ; and commands

    variable declarations

    command block

endproc
```

Ett skript måste ha en huvudprocedur, som inleds med nyckelordet **proc** och avslutas med nyckelordet **endproc**.

Variabler måste deklarerars innan du lägger in kommandon. Först exekveras det första kommandot i huvudproceduren och sedan exekveras de följande kommandona i den ordning de förekommer i skriptet. Skriptet avslutas när det når slutet på huvudproceduren.

### 3.0 Variabler

Skript kan innehålla variabler. Variabelnamn måste börja med en bokstav eller med understreck ("\_") och kan bestå av en sekvens av versaler, gemener, siffror och understreck. Reserverade ord går inte att använda som variabelnamn. För mer information, se listan med reserverade ord i slutet av det här dokumentet.

Du måste deklarerar variabler innan du använder dem. När du deklarerar en variabel så definierar du dess typ. En variabel av en viss typ kan bara innehålla värden av samma typ. Det finns tre typer av variabler:

<u>Typ</u>	<u>Beskrivning</u>
integer	Negativt eller positivt tal, tex 7, -12, och 5698.
string	En sekvens av tecken omgärdade av citattecken, tex, "Hej!" och "Skriv in ditt lösenord:".
boolean	Logisk boolesk variabel som kan vara antingen TRUE eller FALSE.

Variabler tilldelas värden genom tilldelningssatsen:

*variabel = uttryck*

Variabeln tilldelas värdet av uttrycket.

Exempel:

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

boolean bDone = FALSE

string szIP = (getip 2)

set ipaddr szIP
```

### 3.1 Systemvariabler

Systemvariabler bestäms av skriptkommandon eller informationen som du skriver in när du ansluter Fjärranslutning. Systemvariabler är skrivskyddade, vilket betyder att de inte kan ändras i skriptet. Följande systemvariabler existerar:

<u>Namn</u>	<u>Typ</u>	<u>Beskrivning</u>
\$USERID	String	Användaridentifikation för nuvarande anslutning. Variabelns värde bestäms av det angivna namnet i dialogrutan Anslut till i Fjärranslutning.
\$PASSWORD	String	Lösenordet för nuvarande anslutning. Variabelns värde bestäms av det angivna lösenordet i dialogrutan Anslut till i

Fjärranslutning.

\$SUCCESS	Boolean	Variabelns värde bestäms av särskilda kommandon för att markera om kommandona lyckades eller misslyckades. Ett skript kan fatta vidare beslut baserade på variabelns värde.
\$FAILURE	Boolean	Variabelns värde bestäms av särskilda kommandon för att markera om kommandona lyckades eller misslyckades. Ett skript kan fatta vidare beslut baserade på variabelns värde.

Variablerna kan användas där ett uttryck av samma typ används.

```
transmit $USERID
```

är tex ett giltigt kommando eftersom \$USERID är en variabel av typen string.

## 4.0 Strängkonstanter

Skript för Fjärranslutning stöder avbrottssekvenser och caret translations, enligt nedanstående beskrivning.

<u>Sträng</u>	<u>Beskrivning</u>
<i>^char</i>	Caret translation  Om <i>char</i> är ett värde mellan "@" och "_", så är teckensekvensen omvandlad till ett en-byte värde mellan 0 och 31. ^M är tex konverterat till en vagnretur.  Om <i>char</i> är ett värde mellan a och z, så är teckensekvensen omvandlad till ett en-byte värde mellan 0 och 26.  Om <i>char</i> har något annat värde så behandlas inte teckensekvensen på något speciellt sätt.
<cr>	Vagnretur
<lf>	Radmatning
\"	Citattecken
\^	Cirkumflex
\<	Mindre än-tecken
\\	Omvänt snedstreck

Exempel:

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

## 5.0 Uttryck

Ett uttryck är en kombination av operatorer och argument som beräknas till ett resultat. Uttryck kan användas som värden i kommandon.

Ett uttryck kan kombinera variabler, integer, string och booleska värden med unära och binära operatorer enligt nedanstående tabell. Unära operatorer har högst prioritet. Prioriteten på binära operatorer markeras av deras position i tabellen.

Unära operatorer:

<u>Operator</u>	<u>Typ av Operation</u>
-	Negativt tal
!	Ettkomplement

De binära operatorerna är ordnade efter deras prioritet. Operatorer med hög prioritet kommer först i tabellen.

<u>Operator</u>	<u>Typ av Operation</u>	<u>Typbegränsningar</u>
* /	Multiplikativ	Integer
+ -	Additiv	Integer,String (endast+)
< > <= >=	Relation	Integer
== !=	Likhet	Integer, string, boolean
and	Logiskt OCH	Boolean
or	Logiskt ELLER	Boolean

Exempel:

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

## 6.0 Kommentarer

All text som följer efter ett semikolon ignoreras.

Exempel:

```
; this is a comment

transmit "hej"           ; transmit the string "hej"
```

## 7.0 Nyckelord

Nyckelord bestämmer strukturen på skript. Till skillnad från kommandon så utför inte nyckelord en handling. Nyckelorden är beskrivna nedan.

**proc** *namn*

Markerar början på en procedur. Alla skript måste bestå av en huvudprocedur (**proc** main). Exekveringen av ett skript börjar vid huvudprocedurens början och slutar vid slutet av huvudproceduren.

**endproc**

Markerar slutet på en procedur. Fjärranslutning startar PPP eller SLIP när skriptet har kommit fram till huvudprocedurens **endproc**.

**integer** *namn* [ = värde ]

Deklarerar en variabel av typen integer. Du kan ange ett matematiskt uttryck eller en annan variabel för initiera variabeln.

**string** *namn* [ = *värde* ]

Deklarerar en variabel av typen string. Du kan använda bokstäver eller en annan variabel för att initiera variabeln.

**boolean** *namn* [ = *värde* ]

Deklarerar en variabel av typen boolean. Du kan använda ett booleskt uttryck eller en annan variabel för att initiera variabeln.

## 8.0 Kommandon

Alla kommandon är reserverad ord, vilket innebär att du inte kan deklarera en variabel med samma namn som ett kommando. Kommandona är följande:

**delay** *nSeconds*

Gör en paus i *nSeconds* sekunder innan nästa kommando i skriptet exekveras.

Exempel:

```
delay 2      ; pauses for 2 seconds
delay x * 3  ; pauses for x * 3 seconds
```

**getip** *värde*

Väntar på att ta emot en IP-adress från en fjärrdator. Om din Internet service provider retunerar flera IP-adresser i en sträng, så använd *värde*-parametern för att specificera vilken IP-adress som ska användas i skriptet.

Exempel:

```
; get the second IP address
set ipaddr getip 2

; assign the first received IP address to a variable
szAddress = getip
```

**goto** *etikett*

Går till platsen i skriptet som är markerad med *etikett* och fortsätter exekvera de kommandon som följer.

Exempel:

```
waitfor "Prompt>" until 10
if !$SUCCESS then
    goto BailOut ; jumps to BailOut and executes commands
                  ; following it
endif

transmit "bbs^M"
goto End
```

```
BailOut:
    transmit "^M"
```

## halt

Stannar skriptet. Kommandot tar inte bort terminalfönstret. Du måste klicka på Fortsätt för att upprätta en anslutning. Du kan inte återstarta skriptet.

## if *förutsättning* then

*kommandon*

endif

Exekverar sekvensen av *kommandon* om *förutsättning* är TRUE.

Exempel:

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

## label :

Specificerar en plats i skriptet att hoppa till. En etikett (*label*) måste ha ett unikt namn och följa namnkonventionerna som gäller för variabler.

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

Ändrar antalet bitar i en byte som sänds och mottas under sessionen. Antalet bitar kan vara mellan 5 och 8. Om du inte inkluderar det här kommandot så använder Fjärranslutning den inställning som är specificerad för anslutningen.

Exempel:

```
set port databits 7
```

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

Ändrar portens paritetsschema under sessionen. Om du inte inkluderar det här kommandot så använder Fjärranslutning den inställning som är specificerad för anslutningen.

Exempel:

```
set port parity even
```

## set port stopbits 1 | 2

Ändrar portens stoppbitar under sessionen. Antalet kan vara 1 eller 2. Om du inte inkluderar det här kommandot så använder Fjärranslutning den inställning som är specificerad för anslutningen.

Exempel:

```
set port stopbits 2
```

## set screen keyboard on | off

Aktiverar eller avaktiverar inmatning från tangentbordet till skriptets terminalfönster.

Exempel:

```
set screen keyboard on
```

### **set ipaddr *sträng***

Specificerar arbetsstationens IP-adress under sessionen. *Sträng* måste vara utformat som en IP-adress.

Exempel:

```
szIPAddress = "11.543.23.13"
set ipaddr szIPAddress

set ipaddr "11.543.23.13"

set ipaddr getip
```

### **transmit *sträng* [ , raw ]**

Sänder de tecken som är specificerade i *sträng* till fjärrdatorn.

Fjärrdatorn känner igen avbrottssekvenser och caret translations om inte parametern **raw** inkluderas i kommandot. Parametern **raw** är användbar när du överför systemvariablerna \$USERID och \$PASSWORD och användarnamnet eller lösenordet innehåller tecken som annars skulle tolkas som caret sekvenser eller avbrottssekvenser.

Exempel:

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

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

Väntar till datorn mottar en eller flera av de specificerade strängarna från fjärrdatorn. Om inte parametern **matchcase** inkluderas så spelar det ingen roll om parametern *sträng* skrivs med gemener eller versaler.

Om en matchande sträng tagits emot och **then etikett**-parametern används så kommer kommandot hoppa till den plats i skriptet som är utmärkt med *etikett*.

**Until** *tid*-parametern anger det maximala antalet sekunder som datorn väntar på att ta emot strängen innan den exekverar nästa kommando. Utan det här kommandot kan datorn få vänta för alltid.

Om datorn tar emot en av de specificerade strängarna, så sätts systemvariabeln \$SUCCESS till TRUE. Om antalet sekunder specificerat i *tid* överskrids innan strängen är mottagen så sätts systemvariabeln till FALSE.

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

```

**while** *förutsättning* **do**  
     *kommandon*  
**endwhile**

Exekverar sekvensen av *kommandon* till *förutsättning* är FALSE.

Exempel:

```

integer count = 0

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

```

## 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		