

Kommandosprog til opkaldsscript i Netværk via modem

Copyright (c) 1995 Microsoft Corp.

Indholdsfortegnelse

- 1.0 Oversigt
- 2.0 Den grundlæggende struktur i et script
- 3.0 Variabler
 - 3.1 Systemvariabler
- 4.0 Specielle tegn
- 5.0 Udtryk
- 6.0 Kommentarer
- 7.0 Nøgleord
- 8.0 Kommandoer
- 9.0 Reserverede ord

1.0 Oversigt

Mange Internet-udbydere og onlinetjenester kræver, at du indtaster oplysninger manuelt, f.eks. brugernavn og adgangskode, for at oprette forbindelse. Med understøttelse af script i Netværk via modem kan denne proces automatiseres med et script.

Et script er en tekstfil, der indeholder en række kommandoer, parametre og udtryk, som kræves af Internet-udbyderen eller onlinetjenesten for at oprette forbindelsen og benytte tjenesten. Du kan bruge et almindeligt tekstredigeringsprogram, f.eks. Microsoft Notesblok, til at oprette en scriptfil. Når du har oprettet scriptfilen, kan du knytte den til en bestemt forbindelse i Netværk via modem ved hjælp af Scriptværktøj til Netværk via modem.

2.0 Den grundlæggende struktur i et script

En kommando er den grundlæggende instruktion i et script. Nogle kommandoer kræver parametre, der yderligere definerer, hvad kommandoen skal gøre. Et udtryk er en kombination af operatorer og argumenter, som giver et resultat. Udtryk kan bruges som værdier i alle kommandoer. Eksempler på udtryk er aritmetiske udtryk, sammenligninger og tekststrengene.

Grundlæggende ser et script til Netværk via modem ud som følger:

```
;  
; En kommentar begynder med et semikolon og fortsætter  
; til slutningen af linjen.  
;  
  
proc main  
    ; Et script kan have et vilkårligt antal variabler  
    ; og kommandoer  
  
    Erklæring af variabler  
  
    Kommandoer
```

endproc

Et script skal have en hovedprocedure, der angives med nøgleordet **proc main** og senere afsluttes med nøgleordet **endproc**.

Du skal erklære variabler, før du kan tilføje kommandoer. Den første kommando i hovedproceduren udføres først, og derefter udføres de efterfølgende kommandoer i den rækkefølge, hvori de er angivet i scriptet. Scriptet slutter ved afslutningen af hovedproceduren.

3.0 Variabler

Et script kan indeholde variabler. Navne på variabler skal begynde med et bogstav eller et understregningstegn ('_') og kan indeholde både store og små bogstaver, tal og understregningstegn. Du må ikke bruge et reserveret ord som variabelnavn. Yderligere oplysninger om reserverede ord finder du i oversigten over reserverede ord til sidst i dette dokument.

Du skal erklære variabler, før du kan bruge dem. Når du erklærer en variabel, skal du også definere dens type. En variabel af en bestemt type må kun indeholde værdier, der svarer til den pågældende type. Følgende tre typer variabler understøttes:

<u>Type</u>	<u>Beskrivelse</u>
integer	Heltal. Et negativt eller positivt tal, f.eks. 7, -12 eller 5698.
string	Tekststreng. Et antal tegn omgivet af dobbelte anførelstegn, f.eks. "Hallo!" eller "Skriv adgangskode:".
boolean	Boolesk. En logisk værdi, som enten kan være sand eller falsk (TRUE eller FALSE).

Variabler tildeles værdier på følgende måde:

```
variabel = udtryk
```

Variablerne tildeles en værdi, der svarer til værdien af det pågældende udtryk.

Eksempler:

```
integer antal = 5
integer tid = (4 * 3)
integer i

boolean bDone = FALSE

string szIP = (getip 2)

set ipaddr szIP
```

3.1 Systemvariabler

Systemvariabler defineres af scriptkommandoer eller fastlægges af de oplysninger, du angiver, når du opretter en forbindelse i Netværk via modem. Systemvariabler er skrivebeskyttede, dvs. at de kan ikke ændres i scriptet. Systemvariablerne er:

<u>Navn</u>	<u>Type</u>	<u>Beskrivelse</u>
\$USERID	Tekststreng	Bruger-ID for den aktuelle forbindelse. Denne variabel har

		en værdi lig med det brugernavn, der angives i dialogboksen Tilslut til.
\$PASSWORD	Tekststreng	Adgangskoden for den aktuelle forbindelse. Denne variabel har en værdi lig med den adgangskode, der angives i dialogboksen Tilslut til.
\$SUCCESS	Boolesk	Denne variabel tildeles en værdi af visse kommandoer for at angive, om kommandoen lykkedes eller ej. Et script kan foretage valg baseret på værdien af denne variabel.
\$FAILURE	Boolesk	Denne variabel tildeles en værdi af visse kommandoer for at angive, om kommandoen mislykkedes eller ej. Et script kan foretage valg baseret på værdien af denne variabel.

Disse variabler kan kun bruges i udtryk af samme type. F.eks. er

```
transmit $USERID
```

en gyldig kommando, fordi \$USERID er en variabel af typen tekststreng.

4.0 Specielle tegn

Script i Netværk via modem understøtter escape-sekvenser og oversættelse af tegn med cirkumflex foran, som beskrevet nedenfor.

<u>Specialtegn</u>	<u>Beskrivelse</u>
<i>^tegn</i>	Cirkumflex plus tegn Hvis <i>tegn</i> er en værdi mellem '@' og '_', oversættes det hele til én byte med en værdi mellem 0 og 31. ^M oversættes f.eks. til retur (Enter). Hvis <i>tegn</i> er en værdi mellem a og z, oversættes det hele til én byte med en værdi mellem 1 og 26. Hvis <i>tegn</i> har en anden værdi end disse, oversættes det ikke til noget andet.
<cr>	Retur (Enter)
<lf>	Ny linje
\"	Dobbelt anførselstegn
\^	Enkelt cirkumflex
\<	Enkelt '<'
\	Omvendt skråstreg

Eksempler:

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

5.0 Udtryk

Et udtryk er en kombination af operatører og argumenter, som tilsammen resulterer i en værdi. Udtryk kan bruges som værdier i en kommando.

Et udtryk kan kombinere en vilkårlig variabel (heltal, tekststreng og boolesk værdi) med operatørene i de følgende tabeller. Alle monadiske operatører (operatører, der virker direkte på én operand) har forrang. Rangfølgen af de binære operatører (operatører, der virker på to operander) vises ved deres placering i tabellen.

Monadiske operatører:

<u>Operator</u>	<u>Operation</u>
-	Minus
!	Komplementærværdien

I følgende tabel er de binære operatører angivet i rangfølge med den højeste rangorden først:

<u>Operatører</u>	<u>Operation</u>	<u>Restriktioner mht. type</u>
* /	Multiplikation	Heltal
+ -	Addition	Heltal, tekststreng (tekststreng kun +)
< > <= >=	Sammenligning	Heltal
== !=	Identiteter	Heltal, tekststreng, booleske værdier
and	Logisk OG	Booleske værdier
or	Logisk ELLER	Booleske værdier

Eksempler:

```
antal = 3 + 5 * 40
transmit "Glædelig" + " jul"
delay 24 / (7 - 1)
```

6.0 Kommentarer

Al tekst på en linje efter et semikolon ignoreres.

Eksempler:

```
; dette er en kommentar
transmit "hallo" ; sender tekststrengen "hallo"
```

7.0 Nøgleord

Nøgleord bruges til at angive strukturen i et script. I modsætning til kommandoer udfører de ikke nogen handling. Nøgleordene er opført nedenfor:

proc *navn*

Angiver begyndelsen af en procedure. Ethvert script skal have en hovedprocedure (**proc main**). Udførelsen af det pågældende script starter ved begyndelsen af hovedproceduren og slutter ved afslutningen af hovedproceduren.

endproc

Angiver afslutningen af en procedure. Når et script er udført til **endproc** for hovedproceduren, starter Netværk via modem PPP eller SLIP.

integer *navn* [= *værdi*]

Erklærer en variabel af typen heltal. Du kan bruge et vilkårligt numerisk udtryk eller en variabel af typen heltal til at initialisere variabelen.

string *navn* [= *værdi*]

Erklærer en variabel af typen tekststreng. Du kan bruge en vilkårlig tekststreng eller variabel af typen tekststreng til at initialisere variabelen.

boolean *navn* [= *værdi*]

Erklærer en boolesk variabel. Du kan bruge et vilkårligt boolesk udtryk eller en boolesk variabel til at initialisere variabelen.

8.0 Kommandoer

Alle kommandoer er reserverede ord, hvilket betyder, at disse navne ikke må anvendes som navne på variable. Kommandoerne er opført nedenfor:

delay *nsekunder*

Stopper udførelsen af scriptet i et antal sekunder svarende til værdien af *nsekunder*, før den næste kommando i scriptet udføres.

Eksempler:

```
delay 2          ; stopper i 2 sekunder
delay x * 3     ; stopper i x * 3 sekunder
```

getip *værdi*

Venter på at modtage en IP-adresse fra fjerncomputeren. Hvis Internet-udbyderen returnerer flere IP-adresser i en tekststreng, kan du bruge parameteren *værdi* til at angive, hvilken IP-adresse scriptet skal benytte.

Eksempler:

```
; benyt den anden IP-adresse
set ipaddr getip 2

; tildel den første IP-adresse til en variabel
szAddress = getip
```

goto *etiket*

Springer til det sted i scriptet, som er identificeret ved *etiket*, og fortsætter med at udføre kommandoerne derfra.

Eksempel:

```
waitfor "Prompt>" until 10
if !$SUCCESS then
    goto BailOut ; springer til BailOut og udfører
                  ; kommandoerne derfra
endif
```

```
transmit "bbs^M"
goto End

BailOut:
transmit "^M"
```

halt

Stopper udførelsen af scriptet. Denne kommando fjerner ikke det sidste dialogboks-vindue. Du skal klikke på Fortsæt for at oprette forbindelsen. Du kan ikke genstarte scriptet.

if betingelse then
kommandoer
endif

Udfører rækken af *kommandoer*, hvis *betingelse* er sand (TRUE).

Eksempel:

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

etiket :

Angiver det sted i scriptet, hvor der skal springes til. En etiket skal være et entydigt navn, som skal følge reglerne for variabelnavne.

set port databits 5 | 6 | 7 | 8

Ændrer antallet af bit i de byte, der sendes og modtages i den aktuelle session. Antallet af bit kan være mellem 5 og 8. Hvis du ikke medtager denne kommando, bruger Netværk via modem de indstillinger, der er angivet for forbindelsen.

Eksempel:

```
set port databits 7
```

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

Ændrer paritetsindstillingen for porten i den aktuelle session. Hvis du ikke medtager denne kommando, bruger Netværk via modem de indstillinger, der er angivet for forbindelsen.

Eksempel:

```
set port parity even
```

set port stopbits 1 | 2

Ændrer antallet af stopbit for porten i den aktuelle session. Dette tal kan være enten 1 eller 2. Hvis du ikke medtager denne kommando, bruger Netværk via modem de indstillinger, der er angivet for forbindelsen.

Eksempel:

```
set port stopbits 2
```

set screen keyboard on | off

Aktiverer eller deaktiverer input fra tastaturet i scriptvinduet.

Eksempel:

```
set screen keyboard on
```

set ipaddr *tekststreng*

Angiver IP-adressen for arbejdsstationen i den aktuelle session. *Tekststreng* skal være en IP-adresse.

Eksempler:

```
szIPAddress = "11.543.23.13"
set ipaddr szIPAddress

set ipaddr "11.543.23.13"

set ipaddr getip
```

transmit *tekststreng* [, **raw**]

Sender den angivne *tekststreng* til fjerncomputeren.

Fjerncomputeren genkender specialtegn (escape-sekvenser og cirkumflex plus tegn), medmindre du angiver parameteren **raw**. Parameteren **raw** kan være nyttig, når du sender systemvariablerne \$USERID og \$PASSWORD, hvis brugernavnet eller adgangskoden indeholder specialtegn, som uden parameteren **raw** ville blive tolket som escape-sekvenser eller cirkumflex plus tegn.

Eksempler:

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

waitfor *tekststreng* [, **matchcase**] [**then etiket** { , *tekststreng* [, **matchcase**] **then etiket** }] [**until sekunder**]

Venter, indtil computeren modtager en eller flere af de angivne tekststreng fra fjerncomputeren. I parameteren *tekststreng* skelnes der ikke mellem store og små bogstaver, medmindre du angiver parameteren **matchcase**.

Hvis en tekststreng svarende til *tekststreng* modtages, og parameteren **then etiket** er angivet, vil denne kommando springe til det sted i scriptfilen, der er angivet med *etiket*.

Med den valgfri parameter **until sekunder** defineres det maksimale antal sekunder, computeren skal vente på at modtage tekststrengen, før den går videre og udfører den næste kommando. Uden denne parameter kan computeren komme til at vente evigt.

Hvis computeren modtager en af de angivne tekststreng, tildeles systemvariablen \$SUCCESS værdien TRUE. I modsat fald tildeles den værdien FALSE, når der er gået et antal sekunder svarende til *sekunder*, uden at tekststrengen er modtaget.

Eksempler:

```
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 *betingelse* **do**
 kommandoer
endwhile

Udfører rækken af *kommandoer*, indtil *betingelse* har værdien FALSE.

Eksempel:

```
integer antal = 0

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

9.0 Reserverede ord

Følgende ord er reserverede og må ikke benyttes som navne på variable:

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	