

Linguaggio procedurale per il supporto alla creazione delle procedure di connessione di Accesso remoto

Copyright (c) 1995 Microsoft Corp.

Sommario

- 1.0 Cenni preliminari
- 2.0 Struttura di base di una procedura
- 3.0 Variabili
 - 3.1 Variabili di sistema
- 4.0 Stringhe di caratteri
- 5.0 Espressioni
- 6.0 Commenti
- 7.0 Parole chiave
- 8.0 Comandi
- 9.0 Parole riservate

1.0 Cenni preliminari

Molti provider Internet e servizi in linea richiedono l'immissione manuale delle informazioni necessarie per stabilire una connessione, quali il nome utente e la password. Con il supporto per le procedure di connessione di Accesso remoto, è possibile scrivere una procedura per automatizzare il processo di connessione.

Una procedura è un file di testo contenente una serie di comandi, parametri ed espressioni richiesti dal provider Internet o dal servizio in linea per stabilire la connessione e utilizzare il servizio. Per creare un file di procedura, è possibile utilizzare un qualsiasi editor di testo, quale il Blocco note. Una volta creato il file, sarà possibile assegnarlo ad una connessione di Accesso remoto specifica, utilizzando Procedura di connessione di Accesso remoto.

2.0 Struttura di base di una procedura

I comandi costituiscono le istruzioni di base contenute in un file di procedura. Per meglio definire la funzione di alcuni comandi, è necessaria l'indicazione dei parametri. Un'espressione è una combinazione di operatori e argomenti che producono un risultato. Le espressioni, che è possibile utilizzare come valori in qualsiasi comando, possono essere ad esempio di tipo aritmetico, di confronto e di concatenazione di stringhe.

Di seguito è indicato il formato di base di una procedura di Accesso remoto:

```
;  
; Un commento inizia con un punto e virgola e si estende fino alla  
; fine della riga.  
;  
  
proc main  
    ; Una procedura può contenere un numero illimitato di  
    ; variabili e comandi  
  
    dichiarazioni delle variabili
```

```
blocco di comandi
```

```
endproc
```

Un file di procedura deve contenere una procedura principale, racchiusa tra le parole chiave **proc** e **endproc**.

Le variabili devono essere dichiarate prima dei comandi. I comandi vengono eseguiti nell'ordine indicato nella procedura e l'esecuzione termina quando tutti i comandi inclusi nella procedura principale sono stati eseguiti.

3.0 Variabili

Nelle procedure è possibile includere delle variabili. I nomi delle variabili devono iniziare con una lettera o un carattere di sottolineatura ("_") e possono contenere una sequenza di lettere maiuscole o minuscole, cifre e caratteri di sottolineatura. Non è possibile utilizzare una parola riservata come nome di variabile. Per ulteriori informazioni, fare riferimento all'elenco di parole riservate alla fine di questo documento.

Prima di utilizzare le variabili, è necessario dichiararle e specificarne il tipo. Una variabile di un dato tipo potrà infatti contenere solo valori dello stesso tipo. Di seguito sono indicati i tre tipi di variabile supportati:

<u>Tipo</u>	<u>Descrizione</u>
intero	Numero negativo o positivo, ad esempio 7, -12 o 5698.
stringa	Serie di caratteri racchiusi tra virgolette, ad esempio "Immettere la password".
booleano	Valore logico, TRUE o FALSE.

I valori vengono assegnati alle variabili utilizzando la seguente istruzione:

```
variabile = espressione
```

Alla variabile viene assegnato il valore dell'espressione.

Esempi:

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

boolean bDone = FALSE

string szIP = (getip 2)

set ipaddr szIP
```

3.1 Variabili di sistema

Le variabili di sistema vengono impostate dai comandi inclusi nella procedura oppure sono determinate dalle informazioni immesse quando si imposta una connessione di Accesso remoto. Essendo di sola lettura, le variabili di sistema non possono essere modificate all'interno delle procedure. Di seguito viene fornito un elenco delle variabili di sistema.

<u>Nome</u>	<u>Tipo</u>	<u>Descrizione</u>
\$USERID	Stringa	L'identificativo dell'utente per la connessione corrente. La variabile corrisponde al nome dell'utente specificato nella finestra di dialogo Connessione a di Accesso Remoto.

\$PASSWORD	Stringa	Password per la connessione corrente. La variabile corrisponde alla password dell'utente specificata nella finestra di dialogo Connessione a di Accesso Remoto.
\$SUCCESS	Booleano	Variabile impostata da alcuni comandi per indicare se sono stati eseguiti correttamente. L'esecuzione della procedura potrà dipendere dal valore di questa variabile.
\$FAILURE	Booleano	Variabile impostata da alcuni comandi per indicare se sono stati eseguiti correttamente. L'esecuzione della procedura potrà dipendere dal valore di questa variabile.

Queste variabili potranno essere utilizzate quando è richiesta un'espressione dello stesso tipo. Ad esempio,

```
transmit $USERID
```

è un comando valido, in quanto \$USERID è una variabile di tipo stringa.

4.0 Stringhe di caratteri

Le procedure di connessione di Accesso remoto supportano sequenze escape e conversioni dell'accento circonflesso, come descritto di seguito.

<u>Stringa di caratteri</u>	<u>Descrizione</u>
-----------------------------	--------------------

<i>^carattere</i>	Conversione dell'accento circonflesso
-------------------	---------------------------------------

Se *carattere* corrisponde ad un valore tra "@" e "_", la sequenza sarà convertita in un valore di un byte tra 0 e 31. Ad esempio, ^M sarà convertito in un ritorno a capo.

Se *carattere* corrisponde ad un valore tra "a" e "z", la sequenza sarà convertita in un valore di un byte tra 1 e 26.

Se *carattere* corrisponde ad un qualsiasi altro valore, la sequenza non verrà considerata in modo particolare.

<cr>	Ritorno a capo
<lf>	Avanzamento riga
\"	Virgolette doppie
\^	Accento circonflesso singolo
\<	Segno < singolo
\\	Barra rovesciata

Esempi:

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

5.0 Espressioni

Un'espressione è una combinazione di operatori e argomenti che producono un risultato. Le espressioni possono essere utilizzate come valori in qualsiasi comando.

Un'espressione può combinare qualsiasi variabile o intero, stringa o valore booleano con uno qualsiasi degli operatori unari o binari inclusi nella seguente tabella. Gli operatori unari hanno la precedenza sugli altri operatori, mentre la precedenza degli operatori binari è indicata dalla loro posizione nella tabella.

Di seguito sono elencati gli operatori monadici:

<u>Operatore</u>	<u>Tipo di operazione</u>
-	Meno unario
!	Complemento a uno

Nella seguente tabella sono elencati gli operatori binari nell'ordine di precedenza:

<u>Operatori</u>	<u>Operazione</u>	<u>Restrizioni</u>
* /	Moltiplicazione	Interi
+ -	Addizione	Interi Stringhe (solo +)
< > <= >=	Relazione	Interi
== !=	Uguaglianza	Interi, stringhe, variabili booleane
and	AND logico	Variabili booleane
or	OR logico	Variabili booleane

Esempi:

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

6.0 Commenti

Il testo delle righe introdotte da un punto e virgola sarà ignorato.

Esempi:

```
; questo è un commento
transmit "pronto" ; trasmette la stringa "pronto"
```

7.0 Parole chiave

Le parole chiave specificano la struttura della procedura ma, a differenza dei comandi, non eseguono delle operazioni. Di seguito sono elencate le parole chiave disponibili.

proc *nome*

Indica l'inizio della procedura. È necessario che tutte le procedure contengano una procedura principale (**proc** main). L'esecuzione di una procedura inizia dalla procedura principale e termina alla fine della stessa.

endproc

Indica la fine della procedura. Al termine dell'esecuzione dell'istruzione **endproc** relativa alla procedura principale verrà avviato il protocollo PPP o SLIP.

integer *nome* [= *valore*]

Dichiara una variabile di tipo intero. Per inizializzare la variabile, è possibile utilizzare qualsiasi espressione o variabile numerica.

string *nome* [= *valore*]

Dichiara una variabile di tipo stringa. Per inizializzare la variabile, è possibile utilizzare qualsiasi stringa di caratteri o variabile stringa.

boolean *nome* [= *valore*]

Dichiara una variabile di tipo booleano. Per inizializzare la variabile, è possibile utilizzare qualsiasi espressione o variabile booleana.

8.0 Comandi

Tutti i comandi sono parole riservate, quindi non è possibile dichiarare variabili con lo stesso nome dei comandi. Di seguito sono elencati i comandi disponibili.

delay *nSecondi*

Imposta una pausa del numero di secondi specificati in *nSecondi* prima di eseguire il comando successivo della procedura.

Esempi:

```
delay 2      ; imposta una pausa di due secondi
delay x * 3  ; imposta una pausa di x * 3 secondi
```

getip *valore*

Attende l'invio di un indirizzo IP da parte del computer remoto. Se il provider Internet restituisce diversi indirizzi IP in una stringa, utilizzare il parametro *valore* per specificare quale indirizzo IP dovrà essere utilizzato.

Esempi:

```
; utilizza il secondo indirizzo IP
set ipaddr getip 2

; assegna il primo indirizzo IP ricevuto ad una variabile
szAddress = getip
```

goto *etichetta*

Passa alla posizione nella procedura indicata da *etichetta* e continua da quel punto l'esecuzione dei comandi.

Esempio:

```
waitfor "Prompt>" until 10
if !$SUCCESS then
    goto BailOut ; passa a BailOut ed esegue i comandi da quel
                  ; punto in avanti
endif

transmit "bbs^M"
goto End
```

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

halt

Interrompe la procedura, ma non chiude la finestra di dialogo del terminale. Per stabilire la connessione occorre tuttavia fare clic su Continua, poiché la procedura non potrà essere riavviata.

if *condizione* then

comandi

endif

Esegue la serie di *comandi* se la *condizione* è TRUE.

Esempio:

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

etichetta:

Specifica la posizione a cui passare nella procedura. Il nome indicato nell'etichetta dovrà essere univoco e seguire le convenzioni di denominazione delle variabili.

set port databits 5 | 6 | 7 | 8

Cambia il numero di bit trasmessi e ricevuti durante la sessione per ogni byte. È possibile impostare un numero di bit da 5 a 8. Se non si include questo comando, verranno utilizzate le impostazioni delle proprietà relative alla connessione di Accesso Remoto.

Esempio:

```
set port databits 7
```

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

Modifica le impostazioni di parità della porta durante la sessione. Se non si include questo comando, verranno utilizzate le impostazioni delle proprietà relative alla connessione di Accesso Remoto.

Esempio:

```
set port parity even
```

set port stopbits 1 | 2

Modifica il numero dei bit di stop della porta durante la sessione. È possibile impostare il numero 1 o 2. Se non si include questo comando, verranno utilizzate le impostazioni delle proprietà relative alla connessione di Accesso Remoto.

Esempio:

```
set port stopbits 2
```

set screen keyboard on | off

Attiva o disattiva l'input della tastiera nella finestra del terminale della procedura.

Esempio:

```
set screen keyboard on
```

set ipaddr *stringa*

Specifica l'indirizzo IP della workstation per la sessione. Il formato della *stringa* dovrà essere quello di un indirizzo IP.

Esempi:

```
szIPAddress = "11.543.23.13"  
set ipaddr szIPAddress  
  
set ipaddr "11.543.23.13"  
  
set ipaddr getip
```

transmit *stringa* [, **raw**]

Invia al computer remoto i caratteri indicati in *stringa*.

Se non si include il parametro **raw** nel comando, il computer remoto riconoscerà le sequenze escape e le conversioni dell'accento circonflesso. L'inclusione di questo parametro risulta utile quando si trasmettono le variabili di sistema \$USERID e \$PASSWORD e il nome utente o la password contiene sequenze di caratteri che verrebbero altrimenti interpretate come accenti circonflessi o sequenze escape.

Esempi:

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

waitfor *stringa* [, **matchcase**] [**then** *etichetta* { , *stringa* [, **matchcase**] **then** *etichetta* }] [**until** *intervallo*]

Attende l'invio dal computer remoto di una o più stringhe specificate. Il parametro *stringa* non tiene conto delle maiuscole/minuscole, a meno che non si includa il parametro **matchcase**.

Se si riceve una stringa corrispondente e viene utilizzato il parametro **then etichetta**, si passerà alla posizione indicata dall'*etichetta* nel file di procedura.

Il parametro opzionale **until intervallo** definisce il numero massimo di secondi di attesa per il ricevimento della stringa, prima di passare all'esecuzione del comando successivo. Se non si imposta questo parametro, il computer rimarrà in attesa per un periodo indeterminato.

Se si riceve una delle stringhe specificate, la variabile di sistema \$SUCCESS verrà impostata a TRUE. Se tuttavia la stringa non viene ricevuta entro il numero di secondi specificato dal parametro *intervallo*, la variabile di sistema sarà impostata a FALSE.

Esempi:

```
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 *condizione* **do**
 comandi
endwhile

Esegue la serie di *comandi* fino a quando la *condizione* sarà FALSE.

Esempio:

```
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 Parole riservate

Di seguito sono elencate le parole riservate che non potranno essere utilizzate come nomi di variabile.

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	