

Язык макросов подключения для редактора макросов подключения

Copyright (c) 1995-1996 Microsoft Corp.

Содержание

- 1.0 Общие сведения
- 2.0 Структура макроса
- 3.0 Переменные
 - 3.1 Системные переменные
- 4.0 Строковые константы
- 5.0 Выражения
- 6.0 Комментарии
- 7.0 Ключевые слова
- 8.0 Команды
- 9.0 Зарезервированные слова

1.0 Общие сведения

При работе со многими службами доступа к сети Internet и другими интерактивными службами для установки связи необходимо вручную вводить такие сведения, как имя пользователя и пароль. Редактор макросов подключения позволяет написать макрос для автоматизации этого процесса.

Макрос подключения — это текстовый файл, который содержит набор команд, параметров и выражений, запрашиваемых службой доступа к сети Internet или интерактивной службой для установки связи и начала работы со службой. Для создания макроса можно использовать любой текстовый редактор, в том числе блокнот Windows 95. После создания макроса его можно связать с нужным соединением удаленного доступа с помощью редактора макросов подключения.

2.0 Структура макроса

Основными элементами макроса подключения являются команды. Некоторые команды требуют задания параметров, которые уточняют, какое именно действие выполняет команда. В качестве параметров любой команды допускается использование результатов выражений, состоящих из операторов и аргументов. Примерами выражений являются арифметические операции, слияние строк и операции отношения.

Ниже показана стандартная структура макроса подключения:

```
;  
; Строка макроса, начинающаяся с точки с запятой, является  
; комментарием. Комментарии могут занимать несколько строк,  
; но в начале каждой строки должна стоять точка с запятой.  
;  
  
proc main  
    ; Макрос может содержать любое число переменных  
    ; и команд.
```

описания переменных

блок команд

`endproc`

Макрос должен обязательно содержать главную процедуру, которая начинается с ключевого слова **proc** и заканчивается ключевым словом **endproc**.

Описание переменных помещается перед блоком команд. Выполнение макроса начинается с первой команды главной процедуры, после чего выполняются последующие команды в том порядке, как они описаны в макросе. Выполнение макроса заканчивается после выполнения последней команды главной процедуры.

3.0 Переменные

В макросах допускается использование переменных. Имена переменных должны начинаться с буквы или символа подчеркивания ('_') и могут содержать любую последовательность строчных или заглавных букв, цифр или символов подчеркивания. В качестве имен переменных запрещается использовать зарезервированные слова, список которых приведен в конце данного документа.

Перед использованием любой переменной необходимо предварительно описать ее тип данных, то есть указать, какие значения может хранить переменная. В макросах подключения используются три типа переменных:

<u>Тип</u>	<u>Описание</u>
integer	Отрицательные или положительные целые числа, например 7, -12 или 5698.
string	Строки символов, заключенные в кавычки, например "Привет всем!" или "Password:".
boolean	Логические значения TRUE (истина) или FALSE (ложь).

Инструкция присваивания значения переменной выглядит следующим образом:

переменная = выражение

При выполнении этой инструкции переменной присваивается значение указанного выражения.

Примеры:

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

boolean bDone = FALSE

string szIP = "getip 2"

set ipaddr szIP
```

3.1 Системные переменные

Значения системных переменных задаются при выполнении команд макроса подключения или определяются при установке удаленного подключения. Системные переменные доступны только для

чтения, то их изменение в макросе подключения не допускается. Ниже перечислены системные переменные:

<u>Имя</u>	<u>Тип</u>	<u>Описание</u>
\$USERID	String	Имя пользователя для текущего подключения. Пользователь указывает свое имя при создании удаленного доступа.
\$PASSWORD	String	Пароль для текущего подключения. Пароль также задается пользователем при создании подключения удаленного доступа.
\$SUCCESS	Boolean	Некоторые команды присваивают этой переменной значение TRUE (истина), если команда была выполнена успешно. Проверка значения этой переменной позволяет организовать условное выполнение групп команд макроса.
\$FAILURE	Boolean	Некоторые команды присваивают этой переменной значение TRUE (истина), если при возникла ошибка. Проверка значения этой переменной позволяет организовать условное выполнение групп команд макроса.

Эти системные переменные можно использовать в любом выражении подходящего типа. Например, команда

```
transmit $USERID
```

является правильной, так как \$USERID — переменная строкового типа.

4.0 Строковые константы

Редактор макросов подключения поддерживает обработку управляющих последовательностей, записываемых с помощью символов "^", "<" и "\", как описано в данном разделе.

<u>Константа</u>	<u>Описание</u>
^ <i>символ</i>	Обработка сочетания с символом "^" (крышка). Если <i>символ</i> имеет значение от "@" до "_", данное сочетание преобразуется в однобайтовое значение от 0 до 31. Например, ^M преобразуется в символ возврата каретки. Если <i>символ</i> имеет значение от a до z, данное сочетание преобразуется в однобайтовое значение от 1 до 26. Если <i>символ</i> имеет любое другое значение, обработка сочетания не производится.
<cr>	Возврат каретки
<lf>	Перевод строки
\"	Кавычка
\^	Символ "^" (крышка)
\<	Символ "<" (меньше)

\\ Символ "\" (обратная косая черта)

Примеры:

```
transmit "^М"  
transmit "Иванов^М"  
transmit "<cr><lf>"  
waitfor "<cr><lf>"
```

5.0 Выражения

Выражением называется набор операторов и значений, при вычислении которого получается результат. Выражения можно использовать вместо параметров в любой команде макроса.

Выражение может содержать любые переменные, целые числа, строки или логические значения, а также любые унарные или бинарные операторы, перечисленные в следующих таблицах. Все унарные операторы имеют наивысший приоритет. Приоритет бинарных операторов определяется их расположением в таблице.

Унарные операторы:

<u>Оператор</u>	<u>Тип операции</u>
-	Унарный минус (изменение знака)
!	Двоичное дополнение

В следующей таблице перечислены бинарные операторы в порядке убывания приоритета. Сначала описаны операторы с более высоким приоритетом:

<u>Операторы</u>	<u>Операции</u>	<u>Допустимые типы значений</u>
* /	Умножение, деление	Целые числа
+ -	Сложение, вычитание	Строки (только +), целые числа
< > <= >=	Сравнение	Целые числа
== !=	Равенство	Целые числа, строки, логические
and	Логическое И	Логические значения
or	Логическое ИЛИ	Логические значения

Примеры:

```
count = 3 + 5 * 40  
transmit "Привет" + " всем!"  
delay 24 / (7 - 1)
```

6.0 Комментарии

Вся строка текста, начинающаяся с точки с запятой, считается комментарием.

Примеры:

```
; это комментарий  
transmit "Привет" ; передача строки "Привет"
```

7.0 Ключевые слова

Ключевые слова определяют структуру макроса. В отличие от команд они не вызывают каких-либо действий. Ключевые слова перечислены ниже.

proc *имя*

Обозначение начала процедуры. Все макросы должны содержать главную процедуру (**proc main**). Выполнение макроса начинается с первой команды главной процедуры и завершается на последней команде главной процедуры.

endproc

Обозначение конца процедуры. После выполнения инструкции **endproc** главной процедуры подключение удаленного доступа запускает протокол PPP или SLIP.

integer *имя* [= *значение*]

Описание целочисленной переменной. Для присвоения значения переменной можно использовать любое числовое выражение или другую переменную.

string *имя* [= *значение*]

Описание строковой переменной. Для присвоения значения переменной можно использовать любое строковое выражение или другую переменную.

boolean *имя* [= *значение*]

Описание логической переменной. Для присвоения значения переменной можно использовать любое логическое выражение или другую переменную.

8.0 Команды

Все команды являются зарезервированными словами. Это означает, что названия команд запрещается использовать в качестве имен переменных. В данном разделе перечислены команды макросов подключения:

delay *n* *секунд*

Задержка перед выполнением следующей команды макроса. Время ожидания (в секундах) задается значением параметра *n* *секунд*.

Примеры:

```
delay 2      ; задержка на 2 секунды
delay x * 3  ; задержка на x * 3 секунд
```

getip *значение*

Ожидание адреса IP, получаемого от удаленного компьютера. Если служба доступа к Internet возвращает в строке сразу несколько адресов IP, используйте параметр *значение* для выбора нужного адреса IP.

Примеры:

```
; получение второго адреса IP
set ipaddr getip 2

; присвоение переменной первого полученного адреса IP
```

```
szAddress = getip
```

goto метка

Переход к строке макроса, начинающейся с *метки*, и продолжение выполнения всех описанных далее команд.

Пример:

```
waitfor "Prompt>" until 10
if !$SUCCESS then
    goto BailOut ; переход к метке BailOut и выполнение
                  ; всех команд после нее.
endif

transmit "bbs^M"
goto End

BailOut:
transmit "^M"
```

halt

Остановка выполнения макроса. Данная команда не закрывает диалоговое окно терминала. Для установки связи необходимо нажать кнопку "Продолжить". Перезапуск макроса подключения также невозможен.

if условие then

команды

endif

Выполнение набора *команд*, если *условие* истинно (TRUE).

Пример:

```
if $USERID == "Иванов" then
    transmit "Мой пароль^M"
endif
```

метка :

Определяет место в макросе для выполнения переходов. Метка должна иметь уникальное имя и удовлетворять общим соглашениям для имен переменных.

set port databits 5 | 6 | 7 | 8

Изменение числа бит в байтах, передаваемых и получаемых во время сеанса связи. Число бит может быть от 5 до 8. Если макрос не содержит этой команды, при удаленном доступе будут использованы стандартные значения свойств связи.

Пример:

```
set port databits 7
```

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

Изменение режима проверки четности для порта во время сеанса связи. Если макрос не содержит этой команды, при удаленном доступе будут использованы стандартные значения свойств связи.

Пример:

```
set port parity even
```

set port stopbits 1 | 2

Изменение числа стоповых бит для порта во время сеанса связи. Допускается только два значения: 1 или 2. Если макрос не содержит этой команды, при удаленном доступе будут использованы стандартные значения свойств связи.

Пример:

```
set port stopbits 2
```

set screen keyboard on | off

Разрешение или запрет ввода с клавиатуры в окно терминала.

Пример:

```
set screen keyboard on
```

set ipaddr строка

Определение адреса IP рабочей станции во время сеанса связи. *Строка* должна быть записана в формате адреса IP.

Примеры:

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

transmit строка [, raw]

Отправка *строки* удаленному компьютеру.

Если не указан параметр **raw**, удаленный компьютер будет проводить обработку управляющих команд и сочетаний символов со знаком "крышка" (^). Параметр **raw** полезен при передаче системных переменных \$USERID и \$PASSWORD, когда в имени пользователя или пароле содержится специальная последовательность символов, которая при отсутствии параметра **raw** будет преобразована по обычным правилам.

Примеры:

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

```
waitfor строка [ , matchcase ] [ then метка  
{ , string [ , matchcase ] then метка } ]  
[ until время ]
```

Ожидание получения компьютером одной или нескольких строк от удаленного компьютера. Если не указан параметр **matchcase**, регистр символов в параметре *строка* не учитывается.

Если указан параметр **then метка**, после получения нужной строки выполнение макроса будет продолжено с команды, стоящей после *метки*.

Необязательный параметр **until время** определяет максимальное время в секундах, в течение которого компьютер ожидает получения строки перед выполнением следующей команды. При отсутствии этого параметра время ожидания не будет ограничено, и компьютер будет находиться в состоянии ожидания до получения нужной строки.

После получения компьютером одной из указанных строк системной переменной \$SUCCESS присваивается значение TRUE. Если в течение заданного времени строка не получена, системной переменной присваивается значение FALSE.

Примеры:

```
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 условие do  
    команды  
endwhile
```

Выполнение набора *команд* до тех пор, пока *условие* не станет ложным (FALSE).

Пример:

```
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 Зарезервированные слова

Перечисленные ниже слова являются зарезервированными и не могут использоваться в качестве имен переменных.

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		