
ECHO, @

Kontroluje wyświetlanie linii komend w trakcie przetwarzania pliku wsadowego oraz służy do wyświetlania komunikatów.

ECHO ON OFF	OFF powoduje że linie komend nie są wyświetlane, a jedynie ich wyniki. Domyślnie – ON, która powoduje, że każda komenda jest wyświetlana wraz ze znakiem gotowości, a dopiero po niej efekt jej działania.
ECHO	podaje aktualny stan - ON/OFF
ECHO <i>tekst</i>	wyświetla na ekranie podany ciąg literowy. Nie można wyświetlić znaków: <, >,
ECHO. , /\ +	wyświetla pustą linią
@	Analogicznie do ECHO OFF, jednak odwołuje się tylko do linii którą poprzedza.
@ECHO OFF	Polecenie umieszczone zazwyczaj na początku pliku wsadowego.

REM

Komenda ignorowana. Pozwala na umieszczenie komentarzy w pliku wsadowym

REM *komentarz*

PAUSE

Przerywa wykonywanie programu wsadowego do momentu naciśnięcia dowolnego klawisza. Po komendzie można umieścić tekst wyświetlany podczas wykonywania na ekranie.

PAUSE *komentarz*
PAUSE > NUL Nie wyświetla żadnej informacji.

CALL

Wywołuje inny plik wsadowy, a po jego zakończeniu powraca do macierzystego. Program może też wywoływać samego siebie – powstaje pętla – w takim wypadku należy pamiętać o jej zakończeniu.

CALL *dysk:\ścieżka\plik parametry*

Np.: CALL plik.bat

GOTO

Powoduje skok do miejsca oznaczonego podanego etykieta. Przy poleceniu GOTO można pominąć dwukropek poprzedzający nazwę etykiety.

GOTO *etykieta*
...
:etykieta

EXIST

Sprawdza czy dany plik lub katalog istnieje. Stosowana z IF. Parametr \NUL umożliwia sprawdzenie istnienia katalogu. Można stosować negację NOT.

```
IF [NOT] EXIST [plik/katalog] [\NUL] komenda
```

```
IF EXIST c:\nc\nc.exe DEL c:\nc\nc.exe  
IF NOT EXIST c:\katalog\NUL MD c:\katalog
```

IF

Komenda warunkowa. Jeżeli jest spełniony dany warunek, wykonywana jest komenda, jeśli nie jest ona pomijana. Można stosować negację – NOT.

```
IF [NOT] warunek komenda
```

Np.: IF EXIST a:*.* GOTO KONIEC

FOR

Powtarza komendę dla każdego pliku z podanej grupy. Powtarza wykonanie komenda, podczas gdy zmienna przyjmuje kolejno wartości odpowiadające nazwom plików w zbiorze. Nazwą zmiennej może być dowolny znak, z wyjątkiem cyfr.

```
FOR %zmienna IN (zbiór) DO komenda           w linii komend  
FOR %%zmienna IN (zbiór) DO komenda         w pliku wsadowym
```

Np. Aby wydrukować wszystkie pliki .prn:

```
FOR %%i IN (C:\*.PRN) DO PRINT %%i
```

Sprawdzanie czy katalogi istnieją. Jeśli nie – są tworzone:

```
FOR %%i IN (Jeden Dwa Trzy) DO IF NOT EXIST %%i\NUL MKDIR %%i
```

Dodatkowa funkcja:

Jeżeli określenie zbioru zostanie poprzedzone znakiem „/” to pierwszy znak będzie pierwszą wartością zmiennej, reszta – drugą.

```
FOR %%z IN (/TEMP%) do ECHO %%z
```

Komenda wypisze symbol dysku na jakim znajduje się katalog tymczasowy, a następnie ścieżkę do tego katalogu.

CHOICE

Pozwala na interaktywną komunikację z użytkownikiem.

Wybór użytkownika zostaje przekazany dalej jako ERRORLEVEL. Pierwsza litera ERRORLEVEL=1, druga =2 itd. Pozostałe klawisze powodują tylko wydanie krótkiego dźwięku. Kombinacje CTRL+C lub CTRL+BREAK – ERRORLEVEL=0. Jeśli wykonywanie komendy zostało przerwane w inny sposób, sygnalizuje to ERRORLEVEL=255. Kolejne linie muszą sprawdzać warunek ERRORLEVEL w kolejności od najwyższego do najniższego.

```
CHOICE /C:klawisze tekst /N /S /T:k,nn
```

/C:YNA	- możliwość wyboru klawiszy tu: Y, N, A
/N	- nie wyświetla informacji o klawiszach jakie należy wcisnąć.
/S	- brana pod uwagę jest wielkość liter
/T:k,nn	- po nn sekundach (max = 99) zostanie przyjęte że został wciśnięty klawisz k.

Np.:

```
CHOICE /C:JDT Jeden Dwa Trzy /N /T:J,20
IF ERRORLEVEL 3 GOTO TRZY
IF ERRORLEVEL 2 GOTO DWA
IF ERRORLEVEL 1 GOTO JEDEN
```

ERRORLEVEL

Niektóre polecenia generują wartość ERRORLEVEL która w dalszym toku może być sprawdzana i na tej podstawie zapadają decyzje o dalszym toku wykonywania programu.

ERRORLEVEL przyjmuje wartość liczbową z zakresu 0 – 255.

Wartość ta może być sprawdzana poleceniem IF.

Zawsze należy sprawdzać od najwyższej możliwej wartości.

```
IF ERRORLEVEL 3 Komenda
IF ERRORLEVEL 2 Komenda
IF ERRORLEVEL 1 Komenda
```

```
IF ERRORLEVEL => wartość komenda
```

Jeżeli wartość ERRORLEVEL równa się (lub jest większa !) liczbie – wykonaj komendę.

PARAMETRY

Pliki wsadowe mogą być wywoływane z parametrami..

PLIK.BAT *Parametr1 Parametr2 ...*

Np.: Plik.bat AA BBB

IF %1==wartość GOTO etykieta	- Sprawdzenie czy parametr pierwszy %1 równa się danej wartości. Jeżeli plik został wywołany bez parametrów, linia ta spowoduje wypisanie błędu składni.
IF „%1”==”wartość” GOTO etykieta	- Analogicznie do powyższego, tylko brak parametru nie spowoduje błędu.
IF %1X==X komenda	- Jeżeli program został wywołany bez parametrów, wykonywana jest komenda.
IF „%1”==”” komenda	- Jak powyżej
COPY A:\%1 C:\	- Wykorzystanie parametru do operacji

ZMIENNE

SET zmienna=wartość	- Nadawanie zmiennej wartości
ECHO Wartość: %zmienna%	- Odczytywanie
IF %zmienna%==wartość komenda	

SHIFT

Zmienia pozycję parametrów programu wsadowego, co pozwala na stosowanie więcej niż 10 podstawowych. Programy wsadowe obsługują 9 parametrów (od %1 do %9), podawanych przy uruchamianiu, po nazwie pliku oddzielonych od siebie spacją. Nazwa pliku jest oznaczana jako parametr 0 - %0. Po poleceniu SHIFT parametr %0 jest zapominany, jego miejsce zajmuje %1. Natomiast parametrem %1 zostaje dawny %2 itd. (Wartości parametrów przesuwane są w lewo). %9 przyjmuje wartość %10 – nieużywanego ale pamiętanego.

Np.:

Plik.bat bat com exe - Uruchomienie programu z trzema parametrami

:KOPIOWANIE	
If „%1”==”” GOTO END	- Jeżeli nie ma parametru %1, przeskok do etykiety END
COPY A:*.*%1 C:\	- Kopiowanie plików z rozszerzeniem określonym parametrem %1
SHIFT	- Przesunięcie parametrów.
GOTO KOPIOWANIE	
:END	

Program zostaje uruchomiony z trzema parametrami, lecz w kodzie wykorzystywany jest tylko pierwszy - %1. Program kopiuje wszystkie pliki *.bat z dyskietki na dysk C:\ po czym parametry są przesuwane. Teraz parametrem pierwszym %1 jest com. Następuje kopiowanie plików *.com. Następne przesunięcie %1 to exe. Kopiowanie plików *.exe. Znowu przesunięcie . Teraz parametr %1 ma wartość pustą. Druga linia sprawdza ten warunek i kończy program.