

CALL

Call one batch program from another, or call a subroutine.

Syntax

```
CALL [drive:][path]filename [parameters]
```

```
CALL :label [parameters]
```

```
CALL internal_cmd
```

Key:

pathname The batch program to run.

parameters Any command-line arguments.

:label Jump to a label in the current batch script.

internal_cmd Run an [internal](#) command, first expanding any variables in the argument.

CALL a second batch file

The CALL command will launch a new batch file context along with any specified parameters.

When the end of the second batch file is reached (or if EXIT is used), control will return to just after the initial CALL statement.

Example

```
::-----start main.cmd-----
@echo off
SETLOCAL
SET _first_bit=This text won't change
CALL function 10 first
Echo %_number% %_description%

CALL function 15 second
echo %_description% - %_number%

echo Original Variable: %_first_bit%
::-----end main.cmd-----
::
::-----start function.cmd-----
@echo off
SETLOCAL
:: Some random math with %1
SET /a _first_bit=%1 + 25
SET /a _num=%_first_bit% - 10
:: Store %2
SET _descr=[%2]
:: Now the important part, returning the values:
ENDLOCAL& SET _number=%_num%&SET _description=%_descr%
::-----end function.cmd-----
```

CALL a subroutine (:label)

The CALL command will pass control to the statement after the label specified along with any specified parameters.

To exit the subroutine specify GOTO:eof this will transfer control to the end of the current subroutine.

Arguments can be passed either as a simple string or using a variable:

```
CALL MyScript.cmd "1234"
CALL OtherScript.cmd %_MyVariable%
```

A label is defined by a single colon followed by a name. This is the basis of a batch file [function](#).

```
CALL :sub_display 123
ECHO Done
```

```
GOTO :eof

:sub_display
ECHO The result is %1
GOTO :eof
```

At the end of the subroutine, GOTO :eof will return to the position where you used CALL.

Example

```
@ECHO OFF
SETLOCAL
CALL :s_staff SMITH 100
GOTO s_last_bit

:s_staff
ECHO Name is %1
ECHO Rate is %2
GOTO :eof

:s_last_bit
ECHO The end of the script
```

Advanced usage : CALLing internal commands

```
CALL command [command_parameters]
```

CALL can also be used to run any [internal](#) command (SET, ECHO etc) and also expand any environment variables passed on the same line. This is undocumented behaviour, in fact whenever CALL is run without a : prefix, it will always search disk for a batch file/executable called *command* before running the internal command. The effect of this extra disc access is that CALL SET is significantly slower than CALL, its use in loops or with a large number of variables should be avoided.

Example

```
@ECHO off
SETLOCAL
set server1=frodo3
set server2=gandalf4
set server3=ascom5
set server4=last1

::run the Loop for each of the servers
call :loop server1
call :loop server2
call :loop server3
call :loop server4
goto:eof

:loop
set _var=%1
:: Evaluate the server name
CALL SET _result=%%_var%%
echo The server name is %_result%
goto :eof

:s_next_bit
:: continue below

:: Note the line shown in bold has three '%' symbols
:: The CALL will expand this to: SET _result=%server1%
```

Each CALL does one substitution of the variables. (You can also do CALL CALL... for multiple substitutions)

If you CALL an executable or resource kit utility make sure it's available on the machine where the batch will be running, also check you have the latest [versions](#) of any resource kit utilities.

CALL is an [internal](#) command. If [Command Extensions](#) are disabled, the CALL command will not accept batch labels.

"My mother never saw the irony in calling me a son-of-a-bitch" ~ Jack Nicholson

Related:

[Syntax: Functions](#) - How to package blocks of code.

[CMD](#) - can be used to call a subsequent batch and ALWAYS return even if errors occur.

[GOTO](#) - jump to a label or GOTO :eof

[START](#) - Start a separate window to run a specified program or command.

Equivalent bash command (Linux): [. \(source\)](#) - Run a command script in the current shell, [builtin](#) - Run a shell builtin.

Get Your Google Back

Get the Google App and Chrome and
Start Searching. Download Now

www.google.com

AdChoices 



© Copyright SS64.com 1999-2012
Some rights reserved