

Re: del not working in .bat file

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-11/msg01601.html>

- *From:* "Jim Carlock" <anonymous@localhost>
 - *Date:* Mon, 20 Nov 2006 18:57:02 -0500
-

"RB Smissaert" <bartsmissaert@xxxxxxxxxxxxxxxxxxxx> posted...
: When I use the API with the .bat file it runs fine, so maybe I need
: the .bat file and hopefully the API will sort the wait out.

Try using the CALL command inside the batch file. Without the CALL command you run into problems where the next statements do not get executed.

```
@echo off
Call RunSQL.bat
Call Del file.db
```

There's some ways to check the return value of the previous statements as well, although I'm not sure if DEL returns anything.

Typing HELP CALL in a cmd.exe prompt yields...

```
<cmd.exe command="HELP CALL">
Calls one batch program from another.
```

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

batch-parameters Specifies any command-line information required by the batch program.

If Command Extensions are enabled CALL changes as follows:

CALL command now accepts labels as the target of the CALL. The syntax is:

```
CALL :label arguments
```

A new batch file context is created with the specified arguments and control is passed to the statement after the label specified. You must "exit" twice by reaching the end of the batch script file twice. The first time you read the end, control will return to just after the CALL statement. The second time will exit the batch script. Type GOTO /? for a description of the GOTO :EOF extension that will allow you to

Re: del not working in .bat file

"return" from a batch script.

In addition, expansion of batch script argument references (%0, %1, etc.) have been changed as follows:

%* in a batch script refers to all the arguments (e.g. %1 %2 %3 %4 %5 ...)

Substitution of batch parameters (%n) has been enhanced. You can now use the following optional syntax:

- %~1 – expands %1 removing any surrounding quotes ("")
- %~f1 – expands %1 to a fully qualified path name
- %~d1 – expands %1 to a drive letter only
- %~p1 – expands %1 to a path only
- %~n1 – expands %1 to a file name only
- %~x1 – expands %1 to a file extension only
- %~s1 – expanded path contains short names only
- %~a1 – expands %1 to file attributes
- %~t1 – expands %1 to date/time of file
- %~z1 – expands %1 to size of file
- %~\$PATH:1 – searches the directories listed in the PATH environment variable and expands %1 to the fully qualified name of the first one found. If the environment variable name is not defined or the file is not found by the search, then this modifier expands to the empty string

The modifiers can be combined to get compound results:

- %~dp1 – expands %1 to a drive letter and path only
- %~nx1 – expands %1 to a file name and extension only
- %~dp\$PATH:1 – searches the directories listed in the PATH environment variable for %1 and expands to the drive letter and path of the first one found.
- %~ftza1 – expands %1 to a DIR like output line

In the above examples %1 and PATH can be replaced by other valid values. The %~ syntax is terminated by a valid argument number. The %~ modifiers may not be used with %* </cmd.exe>

While it states that the CALL command is used to call BATCH files, it also works in calling executables and scripts.

Without using the CALL command to execute executables, the next statements in the batch file get lost (at least when one batch file calls another, and it doesn't hurt – I don't think – to use it to call other files).

For instance try the following:

Re: del not working in .bat file

Re: del not working in .bat file

```
<test.bat>
@echo off
xcopy /v test1.bat test11.bat
test11.bat
xcopy /v test2.bat test22.bat
test22.bat
</test.bat>
```

```
<test1.bat>
@echo off
echo.
echo INSIDE test1.bat
echo.
call test2.bat
echo.
echo INSIDE test1.bat
echo.
</test1.bat>
```

```
<test2.bat>
@echo off
echo.
echo INSIDE test2.bat
echo.
pause
echo.
echo INSIDE test2.bat
echo.
</test2.bat>
```

You'll see that the first xcopy gets executed and then the test11.bat executes, which calls test2.bat and that executes. But when it exits the second xcopy never executes.

Hope this helps.

—

Jim Carlock
Post replies to the group.

.