

Setlocal command in Windows

The setlocal command helps start the process of differentiating environment variables in a batch file. This process will continue until paired with the appropriate endlocal command or the batch file end.

The **setlocal** command helps start the process of differentiating environment variables in a batch file. This process will continue until paired with the appropriate **endlocal** command or the batch file end. To better understand how to use this command, see the example below.

Setlocal command syntax

```
setlocal [enableextensions | disableextensions] [enabledelayedexpansion | di
```

Setlocal command argument

Argument

Describe

enableextensions

Activate the command extensions until connected to the appropriate **endlocal** command, regardless of the settings that were applied before the **setlocal** command was run.

disableextensions

Turn off the command extensions until you connect to the appropriate **endlocal** command, regardless of the settings that were applied before the **setlocal** command was run.

enabledelayedexpansion

Allow to extend the delayed environment variable until it is connected to the appropriate **endlocal** command, regardless of the settings that were applied before the **setlocal** command was run.

disabledelayedexpansion

Disable the extension of the delayed environment variable until it connects to the appropriate **endlocal** command, regardless of the settings that were applied before the **setlocal** command was run.

/?

Show help at the command prompt.

Comment setlocal command

Use the **setlocal** command

1. When you use the **setlocal** command outside the script or batch file, it will be completely invalid.

Change environment variables

1. Use the **setlocal** command to change environment variables when you run a batch file. The environmental changes that will be made after you run the **setlocal** command are local changes to the batch file. The Cmd.exe program will restore previous settings when it is paired with the appropriate **endlocal** command or the batch file end.

Nested commands

1. You can use the integration of multiple **setlocal** or **endlocal** commands in a batch program (ie nested commands).

Check command extensions in batch files

1. The **setlocal** command will set the ERRORLEVEL variable.disabledelayedexpansion}, variable ERRORLEVEL will be set to 0 (zero).Otherwise, it will be set to 1. You can use this information in batch scripts to determine if extensions are available, as in the following example:

```
setlocal enableextensions
verify other 2> nul
if errorlevel 1 echo Unable to enable extensions
```

1. Because cmd does not set the ERRORLEVEL variable when the command extension is disabled, the verify command will initialize ERRORLEVEL to a value other than 0 when you use it with an invalid argument.disabledelayedexpansion} and it does not set the ERRORLEVEL variable to 1, the command extensions will not be available.

The **setlocal** command example

You can differentiate environment variables in a batch file, as shown in the following sample script:

```
rem ***** Begin Comment *****
rem ch??ng tr??nh này b?t ??u các ch??ng tr??nh superapp ch??
ng tr??nh trên m?ng,
?ang ghi l?i các k?t qu? vào m?t t?p tin, và hi?n th? t?p tin
rem in Notepad.
rem ***** End Comment *****
@echo off
setlocal
path = g: programssuperapp;% path%
superapp call> c: superapp.out
endlocal
start notepad c: superapp.out
```

See more:

1. Check and fix hard drive errors with CMD on Windows
2. 21 Command Prompt tricks you may not know
3. How to create disk full BAT file repair Windows 10

You finished reading the article "**Setlocal command in Windows**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
