SETLOCAL

Set options to control the visibility of environment variables in a batch file.

```
Syntax
      SETLOCAL

      SETLOCAL EnableDelayedExpansion

      SETLOCAL EnableExtensions | DisableExtensions

Key
   EnableDelayedExpansion  Expand variables at execution time rather than at parse time.

   EnableExtensions        Attempt to enable Command extensions.

   DisableExtensions       Attempt to disable Command extensions.
```

SETLOCAL on it's own, usually at the start of a batch file, will begin localisation of Environment Variables.

Issuing a SETLOCAL command, the batch script will inherit all current variables from the master environment/session.

Issuing an ENDLOCAL command will restore any environment variables present before the SETLOCAL was issued.

If a batch script does not use SETLOCAL and ENDLOCAL then all variables will be Global, i.e. visible and modifiable by other scripts.

Although global variables are easy to work with they are not good practice - for example if you have several batch scripts dealing with filenames (and these scripts may be CALLing one another), the first script may have a variable called `_filename`, the second script a different variable called `file-name` (a different name to avoid conflicting with the first script) a third script now needs something like `file_name` this quickly becomes very difficult to manage.

With local variables you are free to use the same variable names in multiple batch scripts - there is no conflict because the local variables are not visible to any other script.
Local Variables can be passed from one batch routine to another with the ENDLOCAL command.

EnableDelayedExpansion

Setting EnabledDelayedExpansion will cause each variable to be expanded at execution time rather than at parse time.
EnableDelayedExpansion is Disabled by default.

Overloading a variable:

SETLOCAL can be used more than once in the same batch file so that multiple values can be stored in the same Environment Variable. To keep track of variable definitions, SETLOCAL and ENDLOCAL statements should be paired.

```
@Echo off
SETLOCAL
::Standard commission
Set _Commission=20
Echo Standard commission %_Commission%

::Premium commission
SETLOCAL
Set _Commission=30
Echo Premium commission %_Commission%

::back to Standard commission
ENDLOCAL
Echo %_Commission%
```

ENABLEEXTENSIONS / DISABLEEXTENSIONS

Command Extensions are enabled by default, there is rarely any need to disable them.

If Command Extensions are permanently disabled or if a script is running under the Windows 95 command processor `command.com` then SETLOCAL ENABLEEXTENSIONS will not be able to restore them.

A batch file to warn if command extensions are not available:

```
VERIFY errors 2>nul
SETLOCAL ENABLEEXTENSIONS
IF ERRORLEVEL 1 echo Unable to enable extensions
```

Errors

SETLOCAL will set an ERRORLEVEL if given an argument: It will be zero if one of the two valid arguments is given and one otherwise.

SETLOCAL is an internal command.

"A local shop for local people" - The League Of Gentlemen

Related:

ENDLOCAL - End localisation of environment changes in a batch file.
Syntax: Functions - How to package blocks of code
Powershell: Set-PSdebug -strict - Equivalent to 'Option Explicit' in VB
Equivalent bash command (Linux): readonly - Mark variables/functions as readonly