

# Built-in variables

Global built-in variables reflect the state of Logo. Some of these variables are read only, while changing others have effects on Logo programs. A few variables are actually property lists whose properties control Logo's behavior.

## **:BASE**

Sets the numeric base used when printing numbers.

### **Syntax**

```
MAKE "BASE number  
:BASE
```

### **Description**

BASE is a pre-defined name that determines the base in which numbers are output by Logo. BASE requires an integer between 2 and 36 as its input.

### **Example**

```
MAKE "BASE 2  
PRINT 10  
1010  
MAKE "BASE 16  
PRINT 10  
A  
MAKE "BASE 10
```

## **:CASE**

Makes Logo convert its input to upper case or not.

### **Syntax**

```
MAKE "CASE TRUE or false  
:CASE
```

### **Description**

:CASE causes Logo to convert all input to upper case if set to TRUE, except for words that are enclosed in vertical bars. If :CASE is FALSE, Logo does not convert any input to upper case. When Logo starts, it sets :CASE to TRUE. Instead of setting the variable :CASE, you can also set the CASE property of the ["PREFS](#) object.

### **Example**

```
MAKE "CASE FALSE  
PARSE "lower  
Result: [lower]  
MAKE "CASE TRUE
```

```
PARSE "lower  
Result: [LOWER]
```

## **:CURRENT.GRAPHICS**

Contains the current graphics window name.

### **Syntax**

```
:CURRENT.GRAPHICS
```

### **Description**

:CURRENT.GRAPHICS contains the name of the currently active graphics window. This is the frontmost graphics window which executes the [DRAW](#) and [CLEARSCREEN](#) commands. This variable is read-only.

Use the system variable [:CURRENT.WINDOW](#) to change the current graphics window.

### **Example**

```
:CURRENT.GRAPHICS  
Result: GRAPHICS
```

## **:CURRENT.WINDOW**

Contains the current window name.

### **Syntax**

```
:CURRENT.WINDOW  
MAKE "CURRENT.WINDOW windowname
```

### **Description**

The :CURRENT.WINDOW variable contains the name of the currently active window, where the user can type into. Setting the variable causes the window with the given name to become visible, and the user focus is set to that window. If no known window has the focus, :CURRENT.WINDOW outputs a question mark.

A list of available windows may be obtained with the [EVERY](#) "WINDOW command.

### **Example**

```
MAKE "CURRENT.WINDOW "LISTENER
```

## **:DATADIR**

Reports the user's data directory.

**Syntax**

```
:DATADIR
```

**Description**

:DATADIR is a system variable that contains the user's Logo data directory. On a Windows system, this is the directory "Terrapin Logo" within the user's Documents directory. Setting the variable changes the value.

If a file name starts with the characters "~DATA", Logo substitutes these characters with the contents of the variable :DATADIR. The file name "~DATA/TEST.LOGO", for example, could become "C:\MYDATA\TEXT.LOGO" if DATADIR was set to "C:\MYDATA". See also [:LOGODIR](#).

**Example**

```
:DATADIR  
Result: /C/Users/John/Documents/Terrapin Logo
```

**:DESKTOPDIR**

Reports the desktop directory.

**Syntax**

```
:DESKTOPDIR
```

**Description**

:DESKTOP is a system variable that contains the name of the directory of the PC's desktop. It is read only.

**Example**

```
:DESKTOPDIR  
Result: /C/Users/John/Desktop
```

**:ERROR**

Contains the type of the last runtime error.

**Syntax**

```
:ERROR
```

**Description**

The built-in variable ERROR works together with the [CATCH](#) and [THROW](#) commands. Whenever Logo throws a word with the [THROW](#) command inside the instruction list of a [CATCH](#) command, or Logo throws a runtime error, and that error or word is caught, Logo sets :ERROR to the type of error caught. The [CATCH](#) command sets the variable to the empty list before executing its instruction list.

Use one of the following values to throw a Logo error:

ACCESS	A name or a procedure does not exist
SYNTAX	Syntax error
BREAK	The user clicked the yellow light
STOP	The user clicked the red light. Same as TOPLEVEL, but prints the message Stopped!
TOPLEVEL	Thrown by the TOPLEVEL command; returns to toplevel if not caught.
CONVERSION	Data conversion error
INPUT	Missing inputs to a procedure
OUTPUT	A procedure did not output anything
RESULT	Unexpected result
BADPROC	Malformed procedure definition
BADFORMAL	Bad formal argument list
RUNTIME	General runtime error
NAME	A name is already being used
RECURSION	Too many active procedures
MATH	Math error
IO	I/O error
COMMAND	This command is only valid at toplevel
COMPONENT	A component is outside the window

## **:ERRORETEXT**

Reports the text of the last caught runtime error or THROWN word.

### **Syntax**

```
:ERRORETEXT
```

### **Description**

The built-in variable ERRORETEXT works together with the [CATCH](#) and [THROW](#) commands. Whenever Logo throws a word with the [THROW](#) command inside the instruction list of a [CATCH](#) command, or Logo throws a runtime error, and that error or word is caught, Logo sets ERRORETEXT to the text of the runtime error, or to the textual representation of the THROWN word. The [CATCH](#) command sets the variable to the empty list before executing its instruction list.

### **Example**

```
TO TEST.ERROR
  CATCH "ERROR [1 / 0]
  (PRINT "|Error text:| :ERRORETEXT)
END
TEST.ERROR
Error text: Division by Zero
```

## "LOGO.ENV

Contains a property list of global program and system settings.

### Syntax

```
PLIST "LOGO.ENV
```

### Description

LOGO.ENV contains a property list of program and system settings:

- MINIMAL** Set this property to TRUE to switch the UI of Logo to a minimalistic version that does not confuse users. When the property is TRUE, the number of buttons in the tool bar is reduced to two, the edit and output windows and the toolboxes are hidden, and the menu is stripped down to a bare minimum.
- POSITION** Contains the position of the Logo frame window relative to the top left of the screen as a two-element list. Setting this property does not have any effect on the Mac, because Logo does not have a frame window on the Mac. Setting the value to the empty list restores the position to its value at startup.
- SCREEN** Contains the size of the screen workspace as a two element list [width height]. This property is read only.
- SIZE** Contains the size of the Logo frame window as a two-element list [width height]. Setting this property does not have any effect on the Mac, because Logo does not have a frame window on the Mac. Setting the value to the empty list restores the size to its value at startup.
- VERSION** Contains the Logo version as a number, like e.g. 4.1.

### Example

```
GPROP "LOGO.ENV "SCREEN  
Result: [1846 1200]
```

## :LOGODIR

Reports Logo's startup directory.

### Syntax

```
:LOGODIR
```

### Description

:LOGODIR is a system variable that contains the name of the directory where Logo was launched from. It is read only. See also [:DATADIR](#).

### Example

```
:LOGODIR  
Result: /C/Users/John/Documents/Terrapin Logo
```

## :PICTURE.FORMAT

Controls the format of saved pictures.

### Syntax

```
:PICTURE.FORMAT
```

### Description

The built-in variable PICTURE.FORMAT controls the file format of saved pictures when no file extension is given. It may hold one of the following values:

JPEG	A compressed image format used widely on the Internet. It loses information during compression, so multiple loads and saves of a JPEG image will slowly reduce its quality.
BMP	The Windows bitmap format. The files are large, but no data is lost.
GIF	A picture format used widely on the Internet, with a maximum of 256 colors.
PCX	An old picture format from the early days of PC computing.
TIFF	A file format for storing images, popular among graphic artists, the publishing industry, and both amateur and professional photographers in general.

### Example

```
:PICTURE.FORMAT  
Result: BMP
```

## :PRECISION

Sets the precision in which numbers are printed.

### Syntax

```
MAKE "PRECISION number
```

### Description

PRECISION is a pre-defined name that sets the number of decimal places displayed in Logo calculations. The default value of PRECISION when Logo is loaded is 2, the maximum allowed number is 15. Calculations are always performed using the full value of the number, regardless of the value of PRECISION. Another way to set the precision is to set the PRECISION property of the "PREFS object.

### Example

```
PI  
Result: 3.14  
MAKE "PRECISION 15  
PI  
Result: 3.14159265358979
```

## "PREFS

System-wide properties.

### Syntax

```
GPROP "PREFS "propertyname
PPROP "PREFS "propertyname value
PLIST "PREFS
```

### Description

The PREFS object contains a list of system-wide properties that control the look and feel of Logo. All of these values are restored when Logo starts. To force Logo to ignore the saved data, hold the SHIFT key when Logo starts.

Name	Default value	Description
AUTOSAVE	FALSE	If TRUE, Logo saves your environment into the file <code>Environment.LGZ</code> on exit.
BACKGROUND.COLOR	[255 255 255]	The background color of the Listener and Editor windows.
COMMENT.COLOR	[128 128 128]	The color of Logo comments.
CASE	TRUE	If TRUE, Logo converts all input that is not enclosed in vertical bars or back quotes to upper case. Note that Logo resets this property to FALSE during a <a href="#">RESTART</a> command.
DRAGGABLE.OBJECTS	TRUE	Objects can be dragged and dropped.
ERROR.COLOR	[255 0 0]	The color of Logo error messages.
EXTENDED.PROPERTIES	FALSE	If TRUE, the Bitmap and Turtle properties dialogs are replaced with the standard property dialog editors.
FONT	[Consolas 9 0]	The font for the LISTENER window.
HINTS	TRUE	If FALSE, tooltip hints are not displayed.
INPUT.COLOR	[0 0 0]	The color of user input.
LISTENER.LINES	100	The maximum number of lines that the LISTENER window displays.
NAME.COLOR	[0 128 0]	The color of Logo names.
NUMBER.COLOR	[128 0 0]	The color of Logo numbers.
OUTPUT.COLOR	[128 0 128]	The color of Logo output.
PRIMITIVE.COLOR	[0 0 128]	The color of Logo primitives.
PROCEDURE.COLOR	0 0 255]	The color of Logo user-defined procedures.
PROMPT	>	The prompt that Logo displays.
QUIT.ON.ERROR	FALSE	This property can be handy if you write programs that work as stand-alone programs. If set to TRUE, and a runtime error or any other uncaught error occurs, Logo displays an error message box and quits. Note that Logo resets this property to FALSE during a <a href="#">RESTART</a> command; it is not saved to disk when saving the environment.
RELAXED.SYNTAX	TRUE	If set to TRUE, Logo automatically closes all open brackets at the end of an input line. Note that this setting only affects the Listener window, not editors, or loaded Logo files. Also, you do not need to use the colon (as in <code>:VALUE</code> ) to access the contents of a name. <b>Note that this feature is deprecated, and will not be supported in a future version of Logo.</b>

REPLACE.PRIMITIVE	TRUE	If set to TRUE, you can redefine primitives. Needless to say that this option is very dangerous. Note that Logo resets this property to FALSE during a <a href="#">RESTART</a> command; it is not saved to disk when saving the environment.
ROTATING.BITMAPS	TRUE	If set to FALSE, bitmaps do not rotate when their heading changes.
SHOW.HEADINGS	FALSE	If set to TRUE, Logo displays an arrow with every widget to indicate its heading.
SINGLE.EDIT	TRUE	If set to TRUE, Logo uses a single editor window instead of multiple editor windows.
SMALL.ICONS	FALSE	If set to TRUE, Logo displays smaller icons in its toolbar.
STARTANGLE.HEADING	TRUE	If set to TRUE, Logo sets the initial heading of a freshly dropped widget to its STARTANGLE property.
SYNTAX.HILITING	TRUE	If set to FALSE, Logo does not use the colors defined here to display text.
VERBOSE	FALSE	If set to TRUE, Logo prints messages when loading a file.
VISIBLE.TELL	FALSE	If set to TRUE, Logo draw a small box around every widget on the TELL list.

**Example**

```
GPROP "PREFS "PRECISION
Result: 2
```

**:STANDARD.INPUT**

The number of the input stream.

**Syntax**

```
MAKE "STANDARD.INPUT channel
```

**Description**

STANDARD.INPUT is a pre-defined name that controls the source of the Logo input stream. When Logo starts up, the default value of STANDARD.INPUT is 0, which means that all input into Logo is read from the Listener window. To change the source of the input stream to another device such as a disk file, the device must be opened or created to prepare it for input, and STANDARD.INPUT assigned the channel number with the value that the [OPEN](#) command returned. The channel number for the Output window is 1. To redirect Logo's output stream, use [:STANDARD.OUTPUT](#). See also [OPEN](#) and [CREATE](#).

**Example**

```
:STANDARD.INPUT
Result: 0
```

**:STANDARD.OUTPUT**

The number of the output stream.



### Syntax

```
MAKE "STANDARD.OUTPUT channel
```

### Description

STANDARD.OUTPUT is a pre-defined name that controls the destination of the Logo output stream. When Logo starts up, the default value of STANDARD.OUTPUT is 1, which means that all Logo output is written to the Listener window. To change the destination of the output stream to another device such as a disk file, the device must be opened or created to prepare it for output, and STANDARD.OUTPUT assigned the channel number which the value that the [OPEN](#) command returned. The channel number for the Output window is 1. To redirect Logo's input stream, use [:STANDARD.INPUT](#). See also [OPEN](#) and [CREATE](#).

### Example

```
:STANDARD.OUTPUT  
Result: 0
```

## :TAB

Sets the tab stop position.

### Syntax

```
MAKE "TAB number
```

### Description

The system variable TAB sets the tab stop width used in the Editor and Listener windows. This width is pre-set to 4 at startup. The value is saved when Logo exits and restored when Logo starts. Setting TAB is equivalent to setting the TAB property at the [:PREFS](#) object.

### Example

```
:TAB  
Result: 4
```

## :TRACE

Controls the output of debugging information.

### Syntax

```
MAKE "TRACE number
```

### Description

TRACE is a pre-defined name that allows monitoring of procedure or command line execution. TRACE displays each step of a procedure in the trace window without pausing as it is run. TRACE displays the current step being evaluated as well as the

procedure name, if any, line, and output. TRACE can be assigned a combination of the following values:

1	Procedure calls and the execution of procedures
2	Assignments to values
4	Assignments to properties
8	Garbage collections

## **:USERNAME**

Reports the login user ID.

### **Syntax**

```
:USERNAME
```

### **Description**

:USERNAME is a read-only system variable that contains the name the user used to log on to Windows.

If a file name contains the characters "~USER", Logo substitutes these characters with the contents of the variable :USERNAME. The file name "~USER/TEST.LOGO", for example, could become "Michael\TEXT.LOGO" if the user logged on as "Michael".

### **Example**

```
:USERNAME  
Result: John
```