Variables

From TouchDesigner 088 Wiki

A variable is a value (a text string) with a name. e.g. One built-in variable is named `DESKTOP`, and where you see `var('DESKTOP')` in scripts or parameters, it is replaced with its string value, the path to your Desktop. To see many of the variables, open Dialogs -> Variables. See Variables Dialog.

NOTE: Some system variables are members of the App Class.

To access any variable in TouchDesigner, use the python `var('VARNAME')` expression See `var()` in Td_Module. Variable names are case-sensitive, so `var('EAR')` is a different variable than `var('ear')`.

There are five kinds of variables. Each differs depending on where they are kept and who can change them:

- **System Environment Variables** - They come from the operating system when you start TouchDesigner, and you cannot change them within TouchDesigner. They can be edited in the Windows Control Panel: Control Panel -> System -> Advanced System Settings or Advanced -> Environment Variables.
- **Built-in Variables** - They are built into TouchDesigner like `PI` (3.14159), you can use them anywhere, and you cannot change them.
- **Root Variables** - Root Variables are user-created, live at the root (`/`) and are available anywhere in TouchDesigner.
- **Path Variables** now called **Internal Shortcuts** (see Extensions) - They can be used anywhere in a component to get the path of the component. They are used to make components location-independent.
- **Component Variables** - You create variables in a component, and it can only be used anywhere in that component. These are used less now that with python, component extensions have Properties.
- Operator Variables - They are obsoleted by python, where the operator has specific members that hold local variables.

Tip: Three expressions return information about a variable - if it exists, what type it is and where it is located: `varexists("`*`NAME`*`")`, `vartype("`*`NAME`*`")` and `varpath("`*`NAME`*`")`.

Tscript only: For Tscript equivalents, see `varexists() vartype() varpath() var()`. TouchDesigner's convention for System, Built-In, Root and Operator variables is all upper-case, `$TEMPERATURE` for example. Lower-case is usually used for Script Variables. Mixed-case is often used for other variables - `$NumStudents`, for example.

Contents

- 1 System Environment Variables
- 2 Built-in Variables
- 3 Root Variables
- 4 Path Variables
- 5 Component Variables
- 6 Script Variables (Tscript Only)
- 7 Operator Variables (Tscript Only)
- 8 Operator Variables (node-specific) (Tscript Only)

System Environment Variables

System environment variables are loaded each time TouchDesigner is started and are the variables defined in Microsoft Windows when TouchDesigner starts. You can add/edit/delete these variables by opening the Windows Start -> Control Panel dialog, then opening the System -> Advanced System Settings -> Advanced page and then clicking on the Environment Variables button. You can add environment variables for only the current user using the top section, or for all users on your computer by using the bottom section.

`TEMP` - where TouchDesigner creates some temporary files.

Here are some variables you can set to get different behavior in TouchDesigner:

`TOUCH_WEB_LAUNCH` Set the variable `TOUCH_WEB_LAUNCH` to use select a different web browser to use when TouchDesigner launches a browser for things such as help. e.g set `TOUCH_WEB_LAUNCH` to "`firefox.exe`" to use Firefox for help. (`rvar TOUCH_WEB_LAUNCH firefox.exe`) or set in environment variables.

`TOUCH_EDITOR` - set a path of the external text editor you wish to use with TouchDesigner.

`TOUCH_TABLE_EDITOR` - set to the path of the external table editor you wish to use with TouchDesigner. If it is not defined, the text editor will be used.

`TOUCH_TEXT_CONSOLE` - set to 1 to force the TouchDesigner text console to always open.

`TOUCH_QUICK_CRASH` - disables the saving of a CrashAutoSave.toe and .dmp dump file when TouchDesigner crashes. The confirmation dialog will also be be suppressed.

`TOUCH_DISABLE_CRASH_HANDLERS` - disables all TouchDesigner crash warnings and dialogs.

`TOUCH_FAST_LOAD` - set to force TouchDesigner to start without loading any UI components.

`TOUCH_MIDI_CHAN_OVERRIDE` Setting this variable lets you force MIDI data coming in from either of the two input ports to look like it's a certain MIDI channel number. Set the variable to 0 2 4 forces Input 2 to be mapped to MIDI channel 2, and Output to MIDI Channel 4.

`TOUCH_NO_UPDATE_CHECK` - set to 1 to allow TouchDesigner to startup without checking for software updates.

`TOUCH_START_COMMAND` - can be set to override the default behavior when starting a `.toe` file (ie. will ignore `/start` DATs).

`TOUCH_10_BIT_COLOR` - enables support for 10-bit color, see 10-bit Color Displays for more information. Requires a high color LDC monitor with a DisplayPort connector.

`TOUCH_DOUBLE_DRAW` - TouchDesigner outputs every rendered frame twice. This is useful when rendering a project at half the refresh rate of your displays, as it allows TouchDesigner to reliably perform the frame doubling. Normally, frame doubling is performed by the display and graphics card, and in certain situations these frames can be doubled incorrectly, leading to stuttering. Legacy feature. **Use the parameter in the Window COMP instead of this.**

Built-in Variables

TouchDesigner's Built-in variables are set at startup and cannot be changed or set by the user. You can view built-in variables by opening the Variables Dialog under the **Dialogs** menu and clicking on the **Built-in** tab. You can also list all the Built-in variables using the `bvar` command.

Some built-in variables are:

`AF` - the current frame number of TouchDesigner's internal timeline. (python: `absTime.frame`)

`AT` - the current time in seconds of TouchDesigner's internal timeline. (python: `absTime.seconds`)

`BPM` - the current beats per minute of the local timeline.

`COOKRATE` - the current update rate of TouchDesigner's internal timeline. (python: `cookRate()`)

`CUR_TOUCHBUILD` - the current TouchDesigner build.

`CUR_TOUCHVERSION` - the current TouchDesigner version.

`DESKTOP` and `MYDOCUMENTS` - the path to the user's `Desktop` and `My Documents` directories respectively.

`END` - the current end range of the local timeline.

`F` - the current frame number of the local timeline.

`FPS` - the current frames per second of the local timeline.

`GADGET_DRAGOUT` - the Path to the Panel Component that most recently had its '`dragout`' Panel Value set to 1.

`GADGET_INSIDE` - the path to the panel component that most recently had its '`inside`' panel value set to 1.

`GADGET_ROLLOVER` - the path to the panel component that most recently had its '`rollover`' panel value set to 1.

`GADGET_SELECT` - the path to the panel component that most recently had its '`select`' panel value set to 1.

`USERPATH` - `$HOMEDRIVE/$HOMEPATH/Derivative/TouchDesigner077`, the path to the user's TouchDesigner preference files. It formerly was `$HOME`.

`OP_ROLLOVER` - the path to the node that most recently was rolled over by the cursor in the network editor.

`OP_SELECT` - the path to the node that most recently selected by the cursor in the network editor.

`PANE_SELECT` - the name of the Pane most recently clicked within in the Network Editor.

`PROJECT` - the Path to the Folder containing the current .toe file.

`RF` - the current frame number of the root timeline.

`RT` - the current time value of the root timeline (based on `$RF`).

`RANGESTART` - the current start sub-range of the local timeline.

`RANGEEND` - the current end sub-range of the local timeline.

`SIGNATURE1` - the current first value of the time signature of the local timeline.

`SIGNATURE2` - the current second value of the time signature of the local timeline.

`START` - the current start range of the local timeline.

`SYS_*` - a set of variables describing CPU, GPU versions, memory, driver versions and more.

SYS_MAINMONITOR - returns the index of the main monitor display.

SYS_XRES and SYS_YRES - the system's current monitor resolution in X and Y respectively. However you can get the resolutions of all monitors attached to the computer by using the Monitors DAT. Tscript: monitorw() and monitorh() expressions.

T - the current time value of the local timeline (based on F).

TFS - path to TouchDesigner's installation directory. By default this is C:/Program Files (x86)/Derivative/TouchDesigner088.

TOUCH - path to directory of the .toe file loaded. When opening a new TouchDesigner session by double-clicking the TouchDesigner icon on the desktop, TOUCH will be set to the desktop.

VARNAME - name of the panel value most recently updated.

VAROP - name of the panel component whose panel value most recently updated.

VAROPPATH - path to the panel component whose panel value most recently updated.

VARVALUE - value of the panel value most recently updated.

Root Variables

Root variables are variables which are defined at the top of the network hierarchy (at /, also referred to as 'root'). Being set here allows these variables to be accessed anywhere in the .toe/.tox file, in any network. You can view Root variables by opening the Variables Dialog and clicking on the **Root** tab.

Some examples:

TOENAME - the name of the .toe/.tox file loaded.

TOUCHBUILD - the TouchDesigner build used when the .toe/.tox file was last saved.

TOUCHTIME - the time when the .toe/.tox file was last saved.

TOUCHVERSION - the TouchDesigner version used when the .toe/.tox file was last saved.

Optional variables:

TOUCH_HELP_DELAY - when set to 0, parameter pop-up help is disabled, otherwise it indicates the number of seconds before the pop-up will be displayed.

Tscript: Root variables can also be listed, set, and unset using the rvar Command, which is the legacy 'set' command.

Path Variables

See Internal Shortcut.

Tscript: Every component has a parameter called Path Variable. If you put "body" in the Path Variable, then anywhere inside that component you can put me.var('body') (python) or $body (tscript) and it will expand to be the path of the component, say /project1/character1/body. This is useful for making components location-independent.

Component Variables

Component **variables** are created local to any component, are usable anywhere in the component, and only in that component.

They are usually used with Tscript but can be used with Python, though in Python, properties in Extensions are more flexible.

In Python, you would use the var('VARNAME') expression to fetch a component variable. Setting them in python is difficult, something like op('../../local/set_variables')['varname',1] = value.

In Tscript, use the cvar command to create, list and delete component variables. Component variables for any component can be viewed in the Variables Dialog: Drag the component you want and drop it on the Variables dialog, this will create a tab displaying the component's local variables if any exist.

Component variables are evaluated hierarchically in TouchDesigner's networks. If a component variable is referenced and it is not defined in the local component, then it will search the parent component for the variable. This will continue until the variable is found or it reaches the top of the network hierarchy, /, the root.

How component variables work internally: Component variables are stored inside a Base COMP called local inside the component. The variables are defined inside local in a table DAT named variables. This DAT is fed by a Table DAT called set_variables, which is the DAT that the commands (cvar, rvar) use to insert variables in. This allows tables of other variables to be merged into variables. The **cvar** command can be used to list, set, or unset component variables.

NOTE: Root variables are simply component variables set at the root of the hierarchy, /.

Script Variables (Tscript Only)

Script variables are variables that are set within a script using the set Command. They exist only for the duration of the script.

Operator Variables (Tscript Only)

These variable you can use in operator parameters and scripts. When used in scripts or in the textport, the variables will apply to the current component. For example, if you `cc` to `/project1`, then `$ON` will return `project1` as the name, and `$OPN` will return `root` as the parent name.

`$ON` and `$OPN` - operator's name and parent operator's name respectively. **Tip:** This is the same as the expressions `opname(".")` and `opname("..")`. See `opname()`.

`$OD` and `$OPD` - operator's digits and parents operator's digits respectively. Operator digits are any numeric characters at the end of the operator's name. **Tip:** This is the same as the expressions `Tscript:opdigits(".")|opdigits(".")` and `Tscript:opdigits("..")|opdigits(".")`. See `opdigits()`.

`$OT` - operator type.

`$OF` - operator family.

`$PN` - parameter name.

`$PX` - parameter index.

`$PL` - parameter label.

`$CURRENT` - in the current network, the name of the current node. The Current flag can be set via the network editor or command `opset -c`. Empty string if the node is not a network. same as opcurrent().

`$SELECTED` - in the current network, a list of all nodes that are selected or "picked". The Pick flag can be set via the network editor or command `opset -p`. Empty string if the node is not a network. same as opselect().

`$COMP` - This variable changes depending on which component it's used in. Its defined as `$TOUCH/Comp/`*`current_component_path`*. So for example say you have the node `/geo1/testcomp/file1` (Where `file1` is a File In SOP). And you have `$COMP/mesh.tog` in the File parameter of the File In SOP, it will resolve to `$TOUCH/Comp/geo1/testcomp/mesh.tog`.

Operator Variables (node-specific) (Tscript Only)

Some OP types have variables you can use only in that node's parameters, node-specific Operator Variables. The variables that are available change depending on the OP, you can find out which variables (if any) an OP supports by looking at its help page.

TOPs have `RESX` and `RESY`, the incoming image resolution.

Some of these variables change within the node while it is cooking. These variables change value for each 'unit' that is evaluated. A 'unit' is whatever the operator acts on, so for CHOPs, a 'unit' is a sample, for Point SOP, a 'unit' is a point, and for the Primitive SOP, each 'unit' is a primitive. For example if you use the variable `$TX` in the Point SOP, for each point `$TX` will take on the X position of that point.

Retrieved from "https://www.derivative.ca/wiki088/index.php?title=Variables&oldid=41452"
Category: Touch Glossary