# Scripting Values Table

Last Update: February 7, 2008

This table provides some of the important values that can be extracted for use in PSP scripts.
Values are listed alphabetically, and for each value, the command which provides that value is given. If the value must be extracted from the command return (assigned to the variable Results in this table), the indexing needed to extract that value is also provided.

At the bottom of the table is a section on toggle commands and how to code the parameters for turning the switches unconditionally on and off or for toggling the switches.

I hope this table helps you in your scripting. If you find other values which should be added to the table, please Email me!

| Value | Command | Index |
|---|---|---|
| Alpha Channels, List | ReturnImageInfo | Results['AlphaChannelList'] |
| Alpha Channels, Number | ReturnImageInfo | Results['AlphaNum'] |
| Color | GetMaterial | Results['Color'] |
| Color depth of image | ReturnImageInfo | Results['BitDepth']<br>8 = paletted image<br>24 = flattened 8 bit/channel image<br>32 = unflattened 8 bit/channel image<br>48 = flattened 16 bit/channel image<br>64 = unflattened 16 bit/channel image |
| Colors - Number in layer | CountColors | |
| Colors - Number in image | CountImageColors | |
| Creation Date (PSP files) | ReturnImageInfo | Results['CreatedDate'] |

| | | |
|---|---|---|
| Documents - number open | len(App.Documents) | |
| File name - complete path | ReturnImageInfo | Results['FileName'] |
| File name - complete path (a second way) | App.TargetDocument.Name | |
| File name - complete path (a third way) | App.ActiveDocument.Name | See App.ActiveDocument Note below. |
| File name - title only | App.TargetDocument.Title | |
| File name - title only (a second way) | App.ActiveDocument.Title | See App.ActiveDocument Note below. |
| GetNumber - button clicked | GetNumber | Results['OKButton']<br>0 = cancel<br>1 = OK |
| GetNumber - number entered | GetNumber | Results['EnteredNumber'] |
| GetString - button clicked | GetString | Results['OKButton']<br>0 = cancel<br>1 = OK |
| GetString - string entered | GetString | Results['EnteredText'] |
| Gradient | GetMaterial | Results['Gradient'] |
| Height | ReturnImageInfo | Results['Height'] |
| Height (a second way) | App.TargetDocument.Height | |
| Height (a third way) | App.ActiveDocument.Height | See App.ActiveDocument Note below. |
| Host Version | ScriptProperties() | Results['Host Version'] |
| Image - updated or not | ReturnImageInfo | Results['Modified']<br>0 = no<br>1 = yes |

| | | |
|---|---|---|
| Image - new or previous saved | ReturnImageInfo | Results['Host Version']<br>blank = newly created<br>not blank = previously created |
| Layer, Background | ReturnLayerProperties | Results['IsBackground']<br>0 = no<br>1 = yes |
| Layer, Blend Mode | ReturnLayerProperties | Results['General']['BlendMode'] |
| Layer, Bounding Rectangle<br>Upper left corner<br>Width of rectangle<br>Height of rectangle | ReturnLayerProperties | Results['LayerRect']<br>Results['LayerRect'][0]<br>Results['LayerRect'][1]<br>Results['LayerRect'][2] |
| Layer, Group Link | ReturnLayerProperties | Results['General']['GroupLink']<br>0 = no<br>1 = yes |
| Layer, Highlight Color | ReturnLayerProperties | Results['General']['PaletteHighlightColor'] |
| Layer, Highlight Used | ReturnLayerProperties | Results['General']['UseHighlight']<br>0 = no<br>1 = yes |
| Layer, Is Visible | ReturnLayerProperties | Results['General']['IsVisible']<br>0 = no<br>1 = yes |
| Layer, Link Set | ReturnLayerProperties | Results['General']['LinkSet']<br>0 = no<br>any other number = link set number |
| Layer, Name | ReturnLayerProperties | Results['General']['Name'] |
| Layer, Opacity | ReturnLayerProperties | Results['General'] |

| | | |
|---|---|---|
| | | ['Opacity'] |
| Layer, Transparency Locked | ReturnLayerProperties | Results['General'] ['IsTransparencyLocked']<br>0 = no<br>1 = yes |
| Layer, Type | ReturnLayerProperties | Results['LayerType'] |
| Layers, Number | ReturnImageInfo | Results['LayerNum'] |
| Modified Date (PSP only) | ReturnImageInfo | Results['ModifiedDate'] |
| Modified in this Session | ReturnImageInfo | Results['Modified'] |
| MsgBox - button clicked | MsgBox | Results<br>0 = no, cancel<br>1 = yes, OK |
| Palette color list | ReturnImageInfo | Results['PaletteColorList'] |
| Path to Layer | ReturnLayerProperties | Results['Path'] |
| Pattern | GetMaterial | Results['Pattern'] |
| Resolution | ReturnImageInformation | Results['PixelsPerUnit'] |
| Selection - is there one? | ReturnImageInfo | Results['Selection']<br>0 = no<br>1 = yes |
| Selection - is there one? (another way) | GetRasterSelectionRect | Results['Type'] (not 0 if there is a selection)<br>0 = no<br>1 = yes (floating)<br>2 = yes (non-floating) |
| Selection, rectangle (raster)<br>Upper left corner<br>Width of rectangle<br>Height of rectangle | GetRasterSelectionRect | Results['Rect']<br>Results['Rect'][0]<br>Results['Rect'][1]<br>Results['Rect'][2] |
| Selection, rectangle | GetVectorSelectionRect | Results['VectorRect'] |

| | | |
|---|---|---|
| (vector)<br>Upper left corner<br>Width of rectangle<br>Height of rectangle | | Results['VectorRect'][0]<br>Results['VectorRect'][1]<br>Results['VectorRect'][2] |
| Selection, type | GetRasterSelectionRect | Results['Type']<br>0 = none<br>1 = floating<br>2 = non-floating |
| Size | App.TargetDocument.Size | |
| Size (a second way) | App.ActiveDocument.Size | See App.ActiveDocument Note below. |
| Texture | GetMaterial | Results['Texture'] |
| Units | ReturnImageInformation | Results['Units'] |
| Version | GetVersionInfo | Results['MajorVersion']<br>8 = PSP 8<br>9 = PSP 9<br>10 = PSP X, etc. |
| Version String (full version name) | GetVersionInfo | Results['VersionString']<br>8.10, or 9.01, or 10.03, etc. |
| Width | ReturnImageInfo | Results['Width'] |
| Width (a second way) | App.TargetDocument.Width | |
| Width (a third way) | App.ActiveDocument.Width | See App.ActiveDocument Note below. |

**App.ActiveDocument Note:** In scripts which access many images, do not use App.ActiveDocument - it is completely dynamic, and can give unpredictable results. In these cases, the simplest alternative is to use App.TargetDocument.

## System Date and Time

To extract time and date information from the system, use the following:

```
import time
local = time.localtime (time.time())
```

The date and time information will be returned in a tuple containing 9 integers. For example, the tuple

(2007, 1, 20, 8, 23, 44, 5, 20, 0)

represents January 20, 2007 at 08:23:44, which is a Saturday, the 20th day of the year, and Daylight Savings time is NOT active.

The following code will let you extract each component of the tuple:

local[0] - gives you the 4-digit year (2007 in above example)
str(local[0])[2:] - gives you the last 2 digits of the year (07 in above example)
local[1] - gives you the month (1 in above example)
local[2] - gives you day of the month (20 in above example)
local[3] - gives you the hour (8 in above example)
local[4] - gives you the minute (23 in above example)
local[5] - gives you the second (44 in above example)
local[6] - gives you the weekday, where 0 represents Monday (5 in above example, which represents Saturday)
local[7] - gives you the Julian day of the year (20 in above example)
local[8] - tells you whether you are currently in Daylight Savings time, where 0 indicates "no" and 1 indicates "yes" (0 in above example)

With a little creative coding, you can get the date and time in any format you want. Remember that these are all integers, and as such, can be used in arithmetic operations. To create a date string, such as 1/20/2007, the integers must be changed to strings, like this:

```
str(local[1]) + "/" + str(local[2]) + "/" + str(local[0])
```

To generate that date using a 2-digit year (1/20/07), use the following:

```
str(local[1]) + "/" + str(local[2]) + "/" + str(local[0])[2:]
```

## Toggle Switches

I recently discovered how to control toggle switches in scripts. This information is documented in the Scripting API, but it's a bit difficult to find, and somewhat confusing to read. Therefore, I am repeating this information here.

For toggle switches, a value of 0 turns the toggle on - if the switch is already on, there will be no change. A value of 1 used for one of the toggle switches turns the toggle off - again, if the switch is already off, there will be no change. Finally, a value of 2 for a toggle parameter toggles the state of the switch - if it's on, it will be turned off, and if it's off, it will be turned on.

Toggle parameters are coded like this - the ShowGrid command is used as an example, with the toggle parameter shown in red, turning the Grid ON:

```
# ShowGrid
App.Do( Environment, 'ShowGrid', {
    'ShowGrid': 0,
    'GeneralSettings': {
        'ExecutionMode': App.Constants.ExecutionMode.Default,
        'AutoActionMode': App.Constants.AutoActionMode.Match
        }
    })
```

In that same command, to turn the Grid OFF, code the ShowGrid parameter this way:

```
'ShowGrid': 1,
```

And to toggle the Grid state, code the ShowGrid parameter this way:

```
'ShowGrid': 2,
```

There are other ways to code the toggle parameter. To turn the Grid ON, you can use one of the following:

```
'ShowGrid': App.Constants.Boolean.false,
'ShowGrid': App.Constants.OnOff.On,
'ShowGrid': App.Constants.ShowCommands.Show,
'ShowGrid': App.Constants.EnterCommands.Enter,
'ShowGrid': False,    ### only available in PSP 9 and later versions
```

To turn the Grid OFF, you can use one of these:

```
'ShowGrid': App.Constants.Boolean.true,
'ShowGrid': App.Constants.OnOff.Off,
'ShowGrid': App.Constants.ShowCommands.Hide,
'ShowGrid': App.Constants.EnterCommands.Exit,
'ShowGrid': True,    ### only available in PSP 9 and later versions
```

And to toggle the state of the Grid switch, you can use:

```
'ShowGrid': App.Constants.OnOff.Toggle,
'ShowGrid': App.Constants.ShowCommands.Toggle,
'ShowGrid': App.Constants.EnterCommands.Toggle,
```

The rest of this table lists the commands that control toggle switches, and the parameter used to control the state of the toggle switch. Unless otherwise noted, these commands work in all versions of PSP:

| Command | Toggle Parameter (shown | Restrictions |
|---|---|---|

| | | |
|---|---|---|
| | turned ON) | |
| EditSelection | 'Mode': 0, | |
| HidePalettes | 'Command': 0, | |
| LayerSetVisibility | 'Command': 0, | |
| ScriptSingleStep | 'SingleStep': 0, | Not available in PSP 8 |
| SetObjectVisibility | 'Command': 0, | |
| ShowBrowser | 'Command': 0, | Available in PSP X only |
| ShowBrushVariance | 'ShowBrushVariance': 0, | |
| ShowEffects | 'ShowEffects': 0, | |
| ShowGrid | 'ShowGrid': 0, | In PSP 8, only toggle works |
| ShowGuides | 'ShowGuides': 0, | |
| ShowHistogram | 'Command': 0, | |
| ShowHistoryPalette | 'ShowHistory': 0, | Not available in PSP 8 |
| ShowLayerPalette | 'ShowLayerPalette': 0, | |
| ShowLearningCenter | 'ShowLearningCenter': 0, | |
| ShowMagnifyWindow | 'ShowState': 0, | |
| ShowMaterialPalette | 'Command': 0, | |
| ShowMixerPalette | 'ShowMixerPalette': 0, | Not available in PSP 8 |
| ShowOptionsBar | 'ShowOptionsBar': 0, | |
| ShowOverview | 'Command': 0, | |
| ShowPhotoToolbar | 'ShowPhotoBar': 0, | |
| ShowRulers | 'ShowRulers': 0, | |
| ShowScriptOutput | 'ShowScriptOutput': 0, | |
| ShowScriptToolbar | 'ShowScriptBar': 0, | |
| ShowStandardToolBar | 'ShowStandardToolBar': 0, | |
| ShowStatusBar | 'StatusBarState': 0, | |
| ShowToolsToolBar | 'ShowToolsToolBar': 0, | |
| ShowWebBar | 'ShowWebBar': 0, | |
| SnapToGrid | 'EnableSnapToGrid': 0, | |
| SnapToGuides | 'EnableSnapToGuides': 0, | |
| TabbedDocuments | 'TabbedDocumentState': 0, | Not available in PSP 8 |

About Me ~ Home ~ Email