

Using Operators

Blender's operators are tools for users to access, that can be accessed with Python too which is very useful. Still operators have limitations that can make them cumbersome to script.

The main limits are:

- Can't pass data such as objects, meshes or materials to operate on (operators use the context instead).
- The return value from calling an operator is the success (if it finished or was canceled), in some cases it would be more logical from an API perspective to return the result of the operation.
- Operators' poll function can fail where an API function would raise an exception giving details on exactly why.

Why does an operator's poll fail?

When calling an operator it gives an error like this:

```
>>> bpy.ops.action.clean(threshold=0.001)
RuntimeError: Operator bpy.ops.action.clean.poll() failed, context is incorrect
```

Which raises the question as to what the correct context might be?

Typically operators check for the active area type, a selection or active object they can operate on, but some operators are more strict when they run. In most cases you can figure out what context an operator needs by examining how it's used in Blender and thinking about what it does.

If you're still stuck, unfortunately, the only way to eventually know what is causing the error is to read the source code for the poll function and see what is checking. For Python operators it's not so hard to find the source since it's included with Blender and the source file and line is included in the operator reference docs. Downloading and searching the C code isn't so simple, especially if you're not familiar with the C language but by searching the operator name or description you should be able to find the poll function with no knowledge of C.

Note

Blender does have the functionality for poll functions to describe why they fail, but it's currently not used much, if you're interested to help improve the API feel free to add calls to `bpy.types.Operator.poll_message_set` (CTX_wm_operator_poll_msg_set in C) where it's not obvious why poll fails, e.g:

```
>>> bpy.ops.gpencil.draw()
RuntimeError: Operator bpy.ops.gpencil.draw.poll() Failed to find Grease Pencil data to d
```

The operator still doesn't work!

Certain operators in Blender are only intended for use in a specific context, some operators for example are only called from the properties editor where they check the current material, modifier or constraint.

Examples of this are:

- `bpy.ops.texture.slot_move`
- `bpy.ops.constraint.limitdistance_reset`
- `bpy.ops.object.modifier_copy`
- `bpy.ops.buttons.file_browse`

Another possibility is that you are the first person to attempt to use this operator in a script and some modifications need to be made to the operator to run in a different context. If the operator should logically be able to run but fails when accessed from a script it should be reported to the bug tracker.

