Abstract

Argument Clinic is a preprocessor for CPython C files. Its purpose is to automate all the boilerplate involved with writing argument parsing code for "builtins". This document shows you how to convert your first C function to work with Argument Clinic, and then introduces some advanced topics on Argument Clinic usage.

Currently Argument Clinic is considered internal-only for CPython. Its use is not supported for files outside CPython, and no guarantees are made regarding backwards compatibility for future versions. In other words: if you maintain an external C extension for CPython, you're welcome to experiment with Argument Clinic in your own code. But the version of Argument Clinic that ships with the next version of CPython *could* be totally incompatible and break all your code.

1 The Goals Of Argument Clinic

Argument Clinic's primary goal is to take over responsibility for all argument parsing code inside CPython. This means that, when you convert a function to work with Argument Clinic, that function should no longer do any of its own argument parsing—the code generated by Argument Clinic should be a "black box" to you, where CPython calls in at the top, and your code gets called at the bottom, with `PyObject *args` (and maybe `PyObject *kwargs`) magically converted into the C variables and types you need.

In order for Argument Clinic to accomplish its primary goal, it must be easy to use. Currently, working with CPython's argument parsing library is a chore, requiring maintaining redundant information in a surprising number of places. When you use Argument Clinic, you don't have to repeat yourself.

Obviously, no one would want to use Argument Clinic unless it's solving their problem—and without creating new problems of its own. So it's paramount that Argument Clinic generate correct code. It'd be nice if the code was faster, too, but at the very least it should not introduce a major speed regression. (Eventually Argument Clinic *should* make a major speedup possible—we could rewrite its code generator to produce tailor-made argument parsing code, rather than calling the general-purpose CPython argument parsing library. That would make for the fastest argument parsing possible!)

Additionally, Argument Clinic must be flexible enough to work with any approach to argument parsing. Python has some functions with some very strange parsing behaviors; Argument Clinic's goal is to support all of them.

Finally, the original motivation for Argument Clinic was to provide introspection "signatures" for CPython builtins. It used to be, the introspection query functions would throw an exception if you passed in a builtin. With Argument Clinic, that's a thing of the past!

One idea you should keep in mind, as you work with Argument Clinic: the more information you give it, the better job it'll be able to do. Argument Clinic is admittedly relatively simple right now. But as it evolves it will get more sophisticated, and it should be able to do many interesting and smart things with all the information you give it.

2 Basic Concepts And Usage

Argument Clinic ships with CPython; you'll find it in `Tools/clinic/clinic.py`. If you run that script, specifying a C file as an argument:

```
$ python3 Tools/clinic/clinic.py foo.c
```

Argument Clinic will scan over the file looking for lines that look exactly like this:

```
/*[clinic input]
```

When it finds one, it reads everything up to a line that looks exactly like this:

```
[clinic start generated code]*/
```