

## CoCoALib - Feature #24

### object files collected in one directory

08 Nov 2011 10:11 - Anna Maria Bigatti

<b>Status:</b>	Rejected	<b>Start date:</b>	08 Nov 2011
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Anna Maria Bigatti	<b>% Done:</b>	100%
<b>Category:</b>	Makefile	<b>Estimated time:</b>	1.01 hour
<b>Target version:</b>	CoCoALib-0.99800	<b>Spent time:</b>	1.00 hour
<b>Description</b>			
Step 1: all objects files should be created in one directory. This should facilitate makefile dependencies.			
Step 2: all object files should be in directories "obj-OPT", "obj-DEBUG1",...			
<b>Related issues:</b>			
Related to CoCoALib - Feature #665: Integrate Janet/Pommaret basis code		<b>In Progress</b>	<b>11 Feb 2015</b>

### History

#### #1 - 08 Nov 2011 10:30 - Anna Maria Bigatti

- Status changed from New to Resolved

#### #2 - 08 Nov 2011 10:30 - Anna Maria Bigatti

- Status changed from Resolved to In Progress

#### #3 - 01 Apr 2014 17:35 - Anna Maria Bigatti

- Target version set to CoCoALib-0.99533 Easter14

#### #4 - 04 Apr 2014 17:13 - John Abbott

- Target version changed from CoCoALib-0.99533 Easter14 to CoCoALib-1.0

#### #5 - 13 Feb 2015 20:37 - John Abbott

Here is an example I found on the internet; it uses GNU make's VPATH facility:

```
vpath %.cpp ../src

src = Foo.cpp Bar.cpp
test_src = Main.cpp FooTest.cpp BarTest.cpp

objects = $(patsubst %.cpp,obj/%.o,$(src))
test_objects = $(patsubst %.cpp,obj/%.o,$(test_src))

$(objects): | obj

obj:
    @mkdir -p $@

obj/%.o : %.cpp
    @echo $<
    @$ (CXX) $(CXXFLAGS) -c $< -o $@
```

Posting here to keep this example in an archive.

**NOTE** JAA continues to be a little uneasy about using features present only in GNU make (despite the fact that JAA thinks "make" is pretty horrible)

#### #6 - 13 Feb 2015 20:59 - John Abbott

Right now I do not recall why we think it is a good idea to have all object files in a single directory.

Pro:

- easy to tell which object files are newer than libcocoa.a
- any name clashes between object files ought to be more evident (maybe? not entirely sure)
- some other (open source) projects do it this way

Con:

- not easy to achieve with classical make files

Would it be easy to do this with some other make program (such as cmake)?

#### #7 - 14 Oct 2020 12:03 - John Abbott

Do we really want to overhaul the current Makefile structure?

I do not recall any problems lately, and am loath to delve too much into the esoterics of make.

As I recall, the Makefile for Frobbly does some "esoteric" stuff (*i.e.* I don't know how it works).

Right now, I'd be more inclined to invest effort into moving to cmake rather than to try to make standard make do acrobatics.

**PS** this issue was opened 9 years ago; the last comment was 5 years ago. And there are no substantial comments.

#### #8 - 14 Oct 2020 15:35 - John Abbott

- Status changed from *In Progress* to *Rejected*

- Target version changed from *CoCoALib-1.0* to *CoCoALib-0.99800*

- % Done changed from *40* to *100*

- Estimated time changed from *20.00 h* to *1.01 h*

What we have currently works well enough (most of the time), and has been stable for quite a while (at least 3 years).

If it aint broke....

Let sleeping dogs lie...

Rejected; the original idea was good, but portable realization was tricky.