

---

Subject: Re: Nice ways to compile  
Posted by [mmiller3](#) on Mon, 11 Apr 2005 18:42:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I make save files by creating an idl script that compiles all the codes that my application uses (or at least those that I can remember!). I include `resolve_all` and `resolve_all, /class` for all the classes that I need. Then I do a save and exit. So making a save set is just a command like "idl make\_save\_file.pro" (see below). This has all the elegance of makefile that is maintained by hand, which is to say, very little. I've considered trying to make a preprocessor that creates header files so I can use `makedepend`. If incremental compilation were possible (that is, loading compiled code, instead of having to compile in order to make code available), that would be useful, but with IDL, it doesn't seem neccessary.

Usually I create my `make_save_file.pro`'s from listings of all the \*.pro files in the directories where I know I've got code components for a given application. That makes save files with cruft that is never used, but it hasn't (yet) left me with anything missing.

Mike

```
;; make_save_file.pro
.compile first_file
.compile second_file
.compile third_file
;; ...more here as needed...
.compile last_file

.compile main

resolve_all
resolve_all, class=['med_imageobj', $
    'registered_med_imageobj', $
    'indypet_kinetics', $
    'ipvis_kinetics', $
    'showprogress' $
]

save, /routines, filename='main.sav'

exit
```

---