

Configuration Files

More fully-featured Hugo interpreters have the ability to write to an external file. Games like *Future Boy* use this to save settings; my little joke game uses it to share information between two different compiled games. You could use the system for these reasons or plenty of others!

Along the way, I decided it was worthwhile to write a configuration file manager. For one thing, several of the extensions I wrote had the ability to write to configuration file, and it seemed wasteful to have each of them write to a separate file and unreasonable to expect every author to replace everything so it could be written to one file every time. Furthermore, it's easy to confuse reading from and writing to a configuration file; if you make a new compile that adds a new unexpected value to the configuration file, when it comes to read it, the game often misreads what is supposed to be what.

To this end, I created a configuration file system that stacks what instructions are to be followed and checks for certain values here and there. If something new has been added that messes up the order, it goes, "ok, this isn't right" and throws out the whole configuration file and starts anew.

```
#include "configuration.h"
                                including the configuration file manager

object jukebox_config "Next Day Jukebox v1"
{
    in config_instructions
    name_sum 0 ! we don't need a value here but we need the slot
    first_time 1 ! if you want a pre-game question/menu only the first time
                ! the game is run, put a true value here

    load_info
    {
        play_music = readval
!         self.first_time = readval ! uncomment if you want the setup to
                ! run only the first time ever
    }
    save_info
    {
        writeval play_music
!         writeval self.first_time ! uncomment if you want the setup to
run only
                ! the first time ever
    }
    setup ! This property routine gets called at game start
    {
        if self.first_time
        {
            self.first_time = 0
            print "Do you want to play this game with music? ";
            play_music = YesOrNo
            return true ! returning true will cause InitScreen to be
called
        }
    }
}
```

example configuration file object

So, as you can see, the file-writing and file-reading instructions for your game go in the `load_info` and `save_info` properties. Any pre-game questions go in the `setup` property.