## Expert Status Line Configuration

Sometimes, you might need to support completely different status line behaviors all in one game. For instance, I had to design a way for the Automap Hugo extension (which draws simple ASCII maps in the status line in Glk interpreters) to peacefully coexist with the NewConverse extension (which lists conversation options in the status window), on top of doing, ya know, regular status line stuff. To this end, I created a `printstatuslib` object, which Roodylib checks for children, using their `find_height` and `status_override` properties to determine which instructions should be followed on any given turn.

I won’t go into the specifics of the system just now (it is all somewhat documented in **roodylib.h**), but here is an example `printstatuslib` object:

```
object mapwindow
{
        in printstatuslib
        find_height
        {
                return (call &FindMapHeight)
        }
        draw_window
        {
                return (call &DrawMapWindow)
        }
        status_override 0
}
```