ZScripts Module Programming

June 13, 2013

All modules are loaded from single files, for example, ban_command.js would be a single module. Modules import from other modules as required.

Part I
Hello world!

Make a file called hello_world.js and add the following code:

```
({
    require: [“commands”, “com”, “theme”, “util”]
    ,
    loadModule: function ()
    {
        // This registers our command in the commands database.
        this.commands.registerCommand(“helloworld”, this);
    }
    ,
    helloworld: // this is the command we register!
    {
        desc: “This is a hello world command!”  // This describes the command!
        ,
        aliases: [“hello”] // using /hello should also work!
        ,
        options: { } // leaving this blank for now!
        ,
        perm: function (src) // a function that says if we have permission to use the c
        {
            // noobs shouldn’t be allowed to use this command!
            return sys.name(src) !== “noob”;
        }
        ,
        code: function (src, cmd, chan)
```

```
        {
            this.com.broadcast("Hello world!", this.theme.INFO);
        }
    }
    // we do not need an unloadModule event to unregister the command because the comma
});
```

Now edit main.json and change it to something like this:

```
{"modules":["default","hello_world"]}
```

Reload the server.

Part II
Module Properties

These are the properties that your module should have present in order to integrate correctly.

require

An array of module names to import. Required modules are imported as a property of the module. For example, importing the "com" module will make the functions of the com module available at this.com., for example: this.com.broadcast("A message!");

loadModule

Called when your module is loaded.

unloadModule

Called when your module is unloaded.

Part III
Module methods

onUnloadModule(callback)

Calls the function when the module is unloaded.

Part IV
Script API

loadModule(modname)

The script's loadModule property loads a module. Generally you should avoid calling it manually, instead using the provided main.json file. If the module is already loaded, this function does nothing.

unloadModule(modname)

Unloads the module. If the module is already unloaded, this module does nothing.

reloadModule(modname)

Unloads the module if it is present, then loads it.

registerHandler(handlername, object, propname=handlername)

Registers a handler of the name handlername from object. It uses the object[propname] property, unless propname is omitted. If propname is omitted, handername is used for propname.

For example, script.registerHandler("beforeChatMessage", this);

Part V
Using Common Modules (making commands, etc.)

commands

You will need to require the commands module in order to make your first command. The commands.registerCommand will load an object from your module and put it in the commands database.