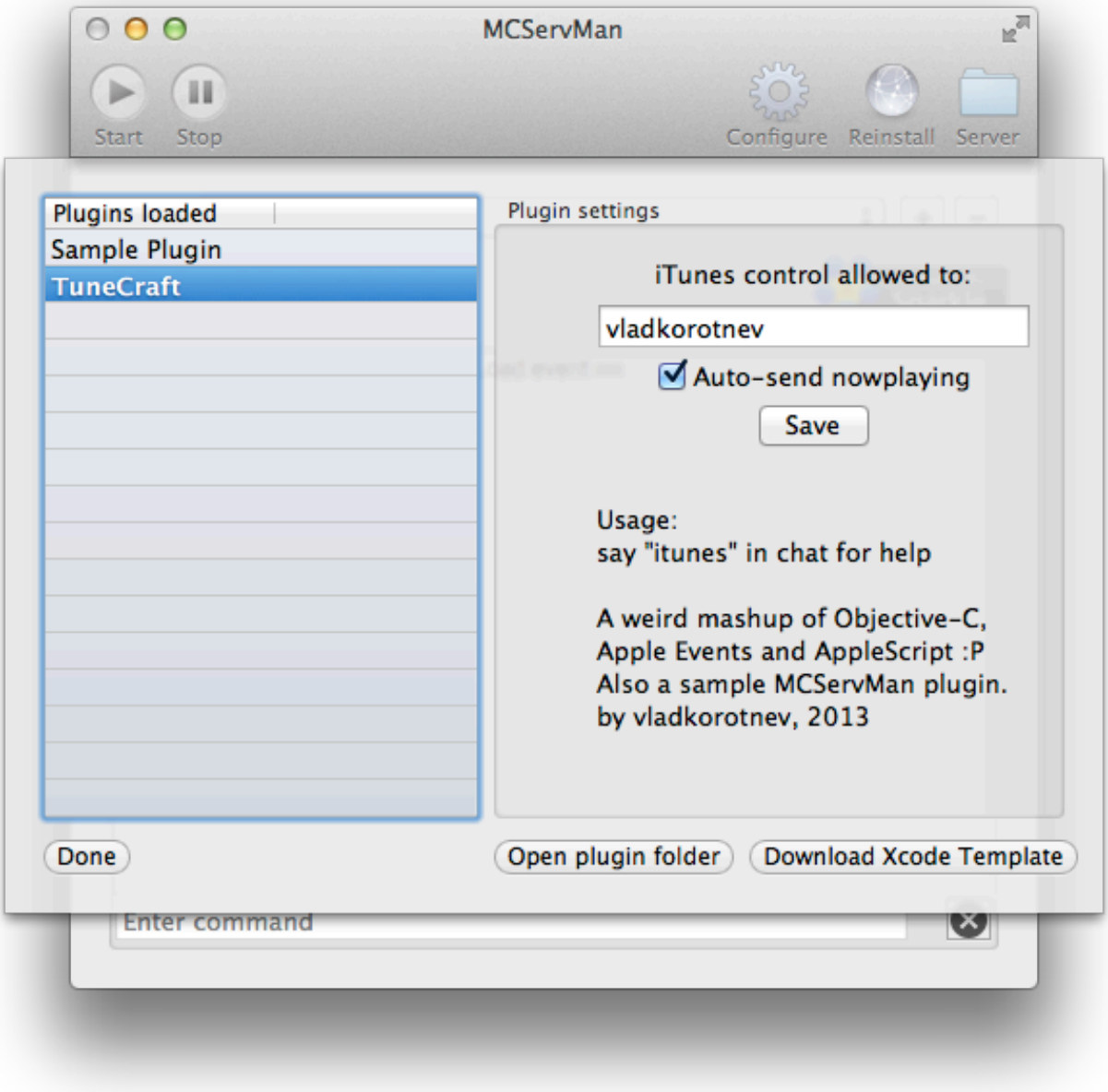
Creating a simple Minecraft Server plugin for MCSM

Step 1 — Install the template

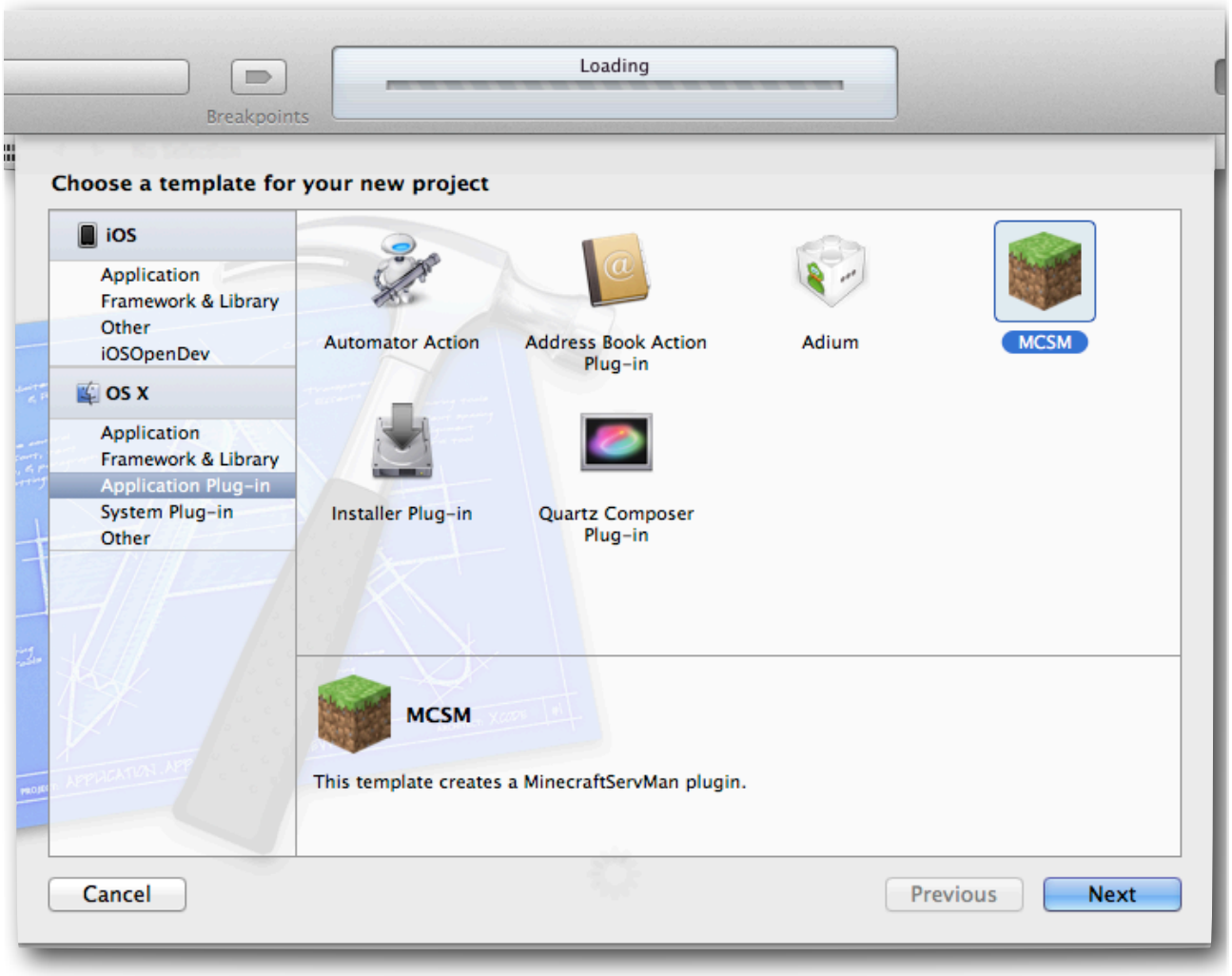
In the plugins panel click the “Download Xcode Template” button



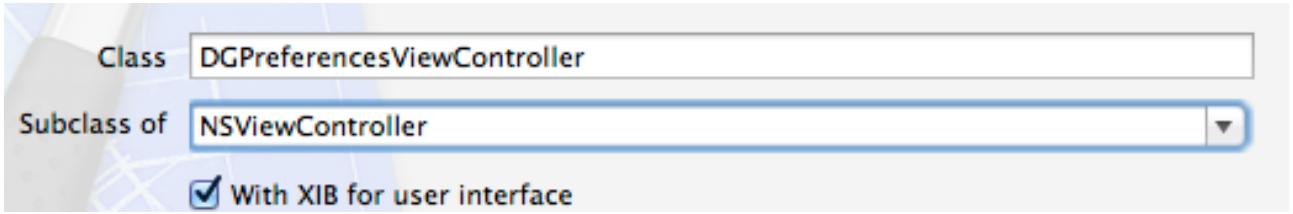
Then run the installer.

Step 2 — Create a new project

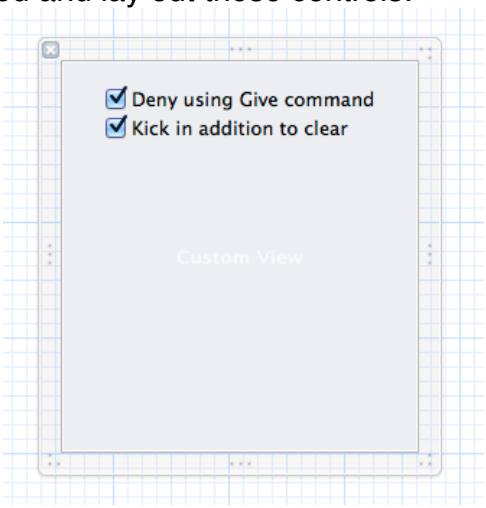
Open Xcode, click “Create a new project”. Select “Application plug-in” and then MCSM.



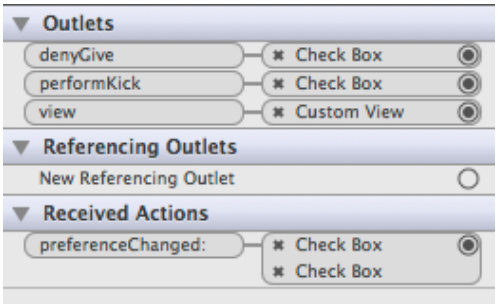
Enter the project name. In this tutorial we will build a plugin that prevents players from using */give*, so enter DoNotGive, for example.
Then select File->New and select “Objective-C class” and enter these settings:



Open the XIB you just created and lay out these controls:



Create and connect respective actions in the ViewController:



Go to the PreferencesViewController.M file and put this code:

```
- (IBAction)preferenceChanged:(id)sender {
    [[NSUserDefaults standardUserDefaults]setBool:self.denyGive.state
forKey:@"DONOTGIVE_denyGive"];
    [[NSUserDefaults standardUserDefaults]setBool:self.performKick.state
forKey:@"DONOTGIVE_kick"];
}
```

In the preferenceChanged method.
Go back into the DoNotGive.m file.
Add this line on the top of the file:

```
#import "DGPreferencesViewController.h"
```

Then add

```
static DGPreferencesViewController*prefs=nil;
static bool isReady=false;
```

after the @implementation line.

Open DGPreferencesViewController.h and add this line before the @end:

```
- (void)refresh;
```

In the respective .m file add:

```
- (void)refresh {
    self.denyGive.state=[[NSUserDefaults
standardUserDefaults]boolForKey:@"DONOTGIVE_denyGive"];
    self.performKick.state=[[NSUserDefaults
standardUserDefaults]boolForKey:@"DONOTGIVE_kick"];
}
```

Then go back to the DoNotGive.m file and change the onSettingsShow method to this:

```
- (void) onSettingShow{
    /* better prepare your viewcontroller to reflect the actual plugin
state here */
    /* called exactly before -settingsView */
    [prefs refresh];
}
```

And uncomment the settingsView method, then change it to this:

```
- (NSView*)settingsView {
   if (prefs == nil) {
        prefs = [[DGPreferencesViewController
alloc]initWithNibName:@"DGPreferencesViewController" bundle:[NSBundle
bundleForClass:self.class]];
    } return prefs.view; }
```

This will, once the user opens settings, create the settings view, update and show it.
Then, after @implementation add:

```
static bool shouldKick=false;
static bool shouldClear=false;
```

to store the preferences in memory instead of reading them all the time from the file.

Change the onServerStart method to update them as well:

```
- (void) onServerStart:
(SMServer<SMServerPluginsAllowedMethodsProtocol>*)server {
    /* the server is now available */
    srv=server; //keep the server connection for further use
    isReady=false;
    shouldKick = [[NSUserDefaults
standardUserDefaults]boolForKey:@"DONOTGIVE_kick"];
    shouldClear = [[NSUserDefaults
standardUserDefaults]boolForKey:@"DONOTGIVE_denyGive"];
}
```

Remove all methods before the @end, except the onServerMessage: and onServerDoneLoading: ones.

In the onServerDoneLoading place the following line:

```
- (void) onServerDoneLoading:
(SMServer<SMServerPluginsAllowedMethodsProtocol>*)server{
    //on server Done loading
    isReady=true;
}
```

Here comes the hardest part — getting who used the give directive.

When the give command is issued, the output may be one of those for Forge:

2013-03-23 14:42:11 [INFO] [Minecraft-Server] [vladkorotnev: Given Lava (ID 10) * 1 to vladkorotnev]
2013-03-23 14:42:36 [INFO] [Minecraft-Server] Given Lava (ID 10) * 1 to vladkorotnev

Or one of those for vanilla server:

2013-03-23 14:45:55 [INFO] [vladkorotnev: Given Lava (ID 10) * 1 to vladkorotnev]
2013-03-23 14:46:32 [INFO] Given Lava (ID 10) * 1 to vladkorotnev

However we will keep the ability for the sysop to give users items, so we'll parse only the first type of the each.

Add this category on top of your DoNotGive.m file for the ease of parsing:

```
@interface NSString (Substr)
- (BOOL)contains:(NSString *)string;
@end

@implementation NSString (Substr)
- (BOOL)contains:(NSString *)string {
    NSRange rng = [self rangeOfString:string options:0];
```



```
    return rng.location != NSNotFound;
}
@end
```

Then, in onServerMessage event, we'll be able to handle either of those:

```
    if ([msg contains:@"[Minecraft-Server]"]) {
        // Parse the Forge line
    } else {
        // Parse the Vanilla line
    }
```

But before that, let's check if we even want that event to be handled:

```
 if(!isReady)return;
if (![msg contains:@"Given"]&&![msg contains:@"to"])return;
```

Yes, it's that easy :)

So, to parse the Forge line, you need to split the line by something repeating. An ideal variant is the *]* character.
Create two variables before the if statement:

```
NSString*event=nil;
NSArray*split=nil;
```

and parse the forge line this way:

```
split = [msg componentsSeparatedByString:@"]"];
event = split[2];
```

This means after those operations, the *split* array will contain these items:
0. 2013-03-23 14:42:11 [INFO
1. [Minecraft-Server
2. [vladkorotnev: Given Lava (ID 10) * 1 to vladkorotnev
3. *<empty item>*

The event variable will then contain the second item of the array;

Do you get how will we parse the Vanilla line? :)
Yes, it has one [enclosed] block less, so we'll change the number of the item to [1], and that's it.

Then, the event variable will contain:
[vladkorotnev: Given Lava (ID 10) * 1 to vladkorotnev

We'll have to split it by " to " (with spaces) to cut the username from there, and store it in a variable:

```
NSString*badPlayer = [event componentsSeparatedByString:@" to "][1];
```

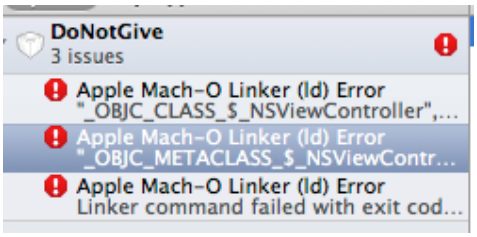
And now, in the end, check settings and do something to the cheater:

```
 if (shouldClear) {
        [srv sendServerMessage:[NSString stringWithFormat:@"clear
%@",badPlayer]];
        if (shouldKick)
            [srv sendServerMessage:[NSString stringWithFormat:@"kick %@
No giving here please",badPlayer]];
        else
```

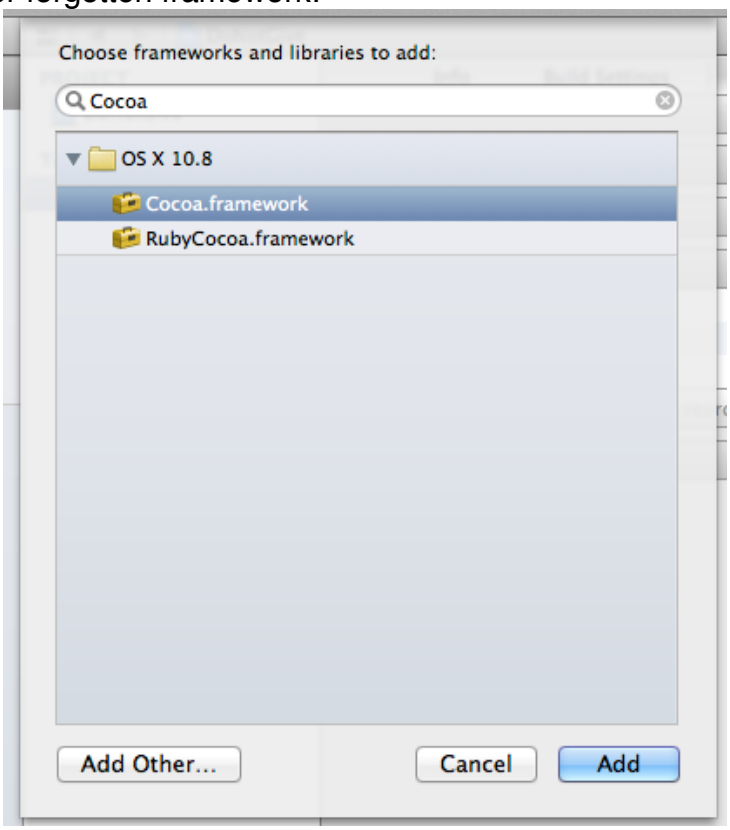


```
            [srv sendServerMessage:[NSString stringWithFormat:@"say
Shame on you, %@",badPlayer]];
}
```

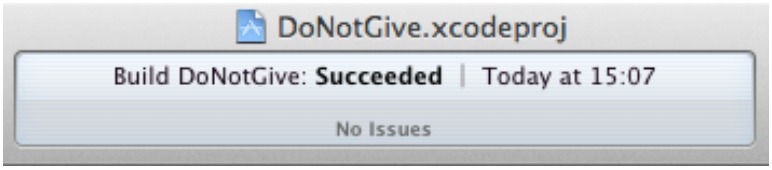
And, you're done! Press Cmd-B to build it.
AAAND SHIT!



Oh well, we forgot to add Cocoa.framework to the project!
Go to the project properties, "Build Phases" tab, the "Link With Libraries" stage, and press the + icon.
Then select the poor forgotten framework:

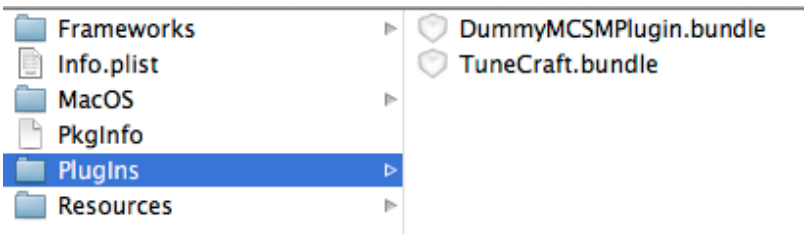


Cmd-B again, and

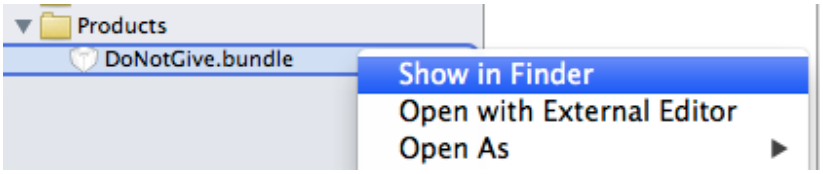


Isn't that cool?

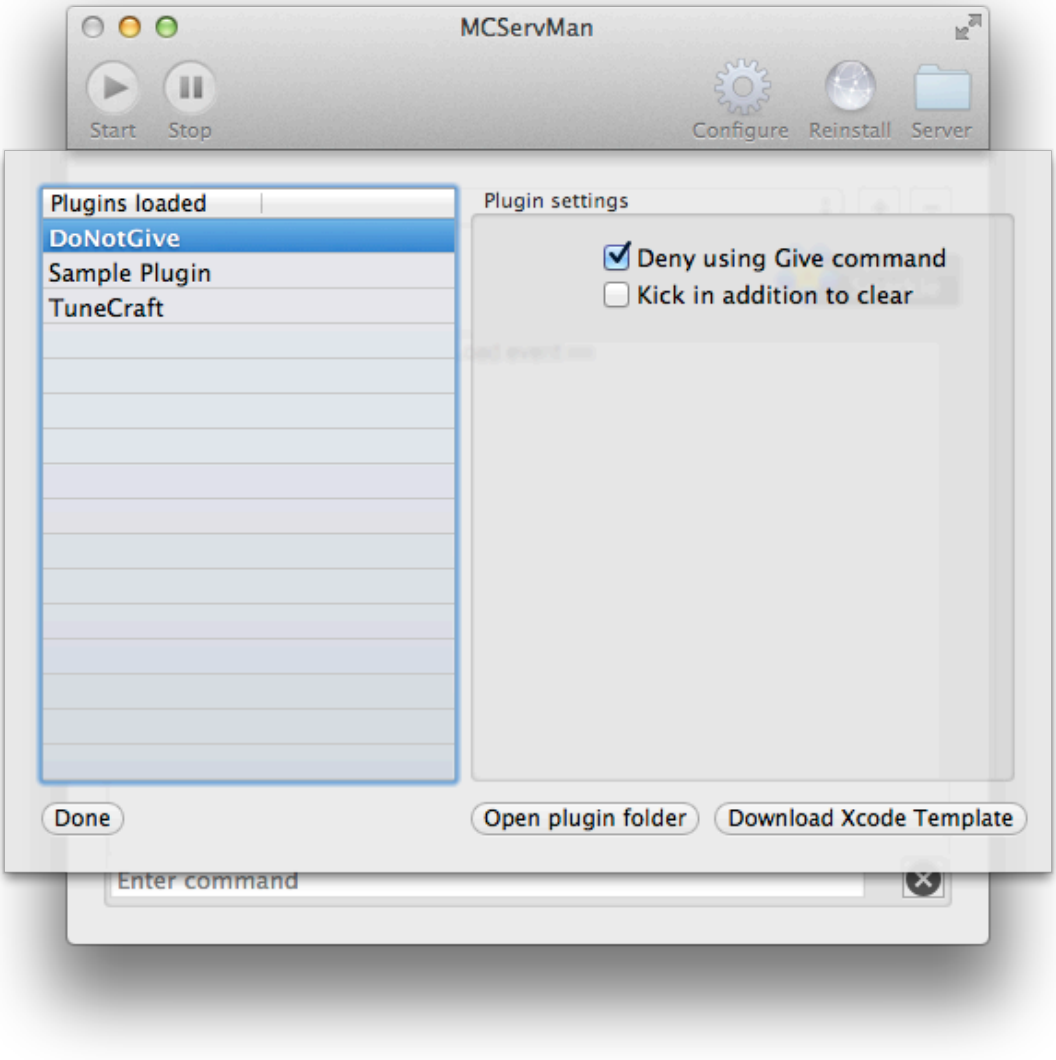
Now open MCServMan, open the Plugin manager, and click "Open plugin folder".



Right-Click the DoNotGive bundle in Xcode and reveal it in Finder:



Put it into the MCSM plugins folder. Quit and reopen MCSM.
Boom!



Log onto your server and give yourself some ~~pie~~ cake.
You will be accused of stealing it!

Or even worse:

Nobody will want to use Give command on your place anymore.
Enjoy your improved Survival server. *(seriously, are you gonna run this on a creative one?)*

If you reached this point and it works the same way, then congratulations, great job!

Complete source code download: http://vladkorotnev.github.com/soft/mcsm/dontgive.zip

