

---

Subject: Re: brenbot plugin needed

Posted by [danpaul88](#) on Mon, 01 Feb 2010 20:57:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Well I am pretty busy with stuff at the moment but after reading this I decided it should be pretty easy to do... simply override the built in auth-command with a plugin one, call the built-in command from the plugin command and then check that the auth was successful.

NOTE: This code is untested as I simply don't have the time right now, I just threw it together in 5 minutes to show how it can be done. It might or might not work as intended.

Oh, just an FYI, the issue with temp mods leaving the game and still being temped when rejoining was fixed several versions ago, in an internal beta prior to 1.50 if I remember correctly.

EDIT: And, before anyone asks, the plugin will only call the attach console command for successful authentications on Admins, Full Mods, Half Mods and Temp Mods, all other successful authentications will NOT trigger the console command.

For anyone interested in the relevant code without downloading the zip file;

Toggle Spoiler# Function for !auth command

```
sub auth
```

```
{  
  my %args = %{$_[ ARG0 ]};
```

```
  # Is the player already authed?
```

```
  my ( $result, %player ) = plugin::getPlayerData($args{'nick'});
```

```
  if ( $result == 1 && (!defined($player{'isAuthed'}) || $player{'isAuthed'} != 1) )
```

```
  {
```

```
    # Call the real auth command
```

```
    commands::auth(@_);
```

```
  # Are they now authed?
```

```
  ( $result, %player ) = plugin::getPlayerData($args{'nick'});
```

```
  if ( $result == 1 && defined($player{'isAuthed'}) && $player{'isAuthed'} == 1 )
```

```
  {
```

```
    # Is the player some type of moderator?
```

```
    if ( plugin::isAdmin($player{'id'}) || plugin::isFullMod($player{'id'}) ||
```

```
plugin::isHalfMod($player{'id'}) || plugin::isTempMod($player{'id'}) )
```

```
    {
```

```
# Attach script
plugin::RenRemCMD('attach $player{id} auth_script');
}
}
}
}
```

In addition to the code for the overridden command you will also need to copy the command declaration from the commands.xml file into the plugins XML file (otherwise the bot won't know your new command exists), add the command event to the plugins events list in the XML and handle the command event in the plugin code in the usual way.

Because plugin commands are looked for first any command in a plugin matching the name of a built in command will override the built in command.

EDIT 2: See 2 posts down for download, fixed a bug. Note that the code in the spoiler above has been updated to the correct version as attached to the post below.

---