Subject: [SSGM Plugin] CheckPlayerName
Posted by roszek on Thu, 14 Aug 2014 00:57:49 GMT

View Forum Message <> Reply to Message

This plugin is a rework of an old plugin of mine that changed spaces from players nicks into underscores when they joined.

It still changes spaces but now it will change any char specified in the ssgm.ini under [InvalidChars] as well. It will also change a player's nick if it contains words listed in ssgm.ini under [UnallowedNicks].

For characters: Add the Characters not wanted in nicks after Invalid=.

For words or names: Add a number in sequence after the word BadWord and then the word or name to look for (BadWord1=someword).

Shown below...

[Plugins]
00=BanSystem.dll
01=Mute.dll
02=CheckPlayerName.dll

[InvalidChars]

Invalid=\.,

[UnallowedNicks]

BadWord1=shit
BadWord2=ass
BadWord3=homo

File Attachments
1) pic1.png, downloaded 919 times

Admin Message from Host

Your nickname contained invalid characters and has been changed from +King Fred to _King_Fred

Okay

2) `pic2.png`, downloaded 908 times

Admin Message from Host

The unallowed word <ass> was found in your nickname. Your nickname has been changed from The_assjack to Player762.

Okay

3) `CheckPlayerName.rar`, downloaded 265 times

Subject: Re: [SSGM Plugin] InvalidChars
Posted by iRANian on Thu, 14 Aug 2014 07:26:20 GMT
View Forum Message <> Reply to Message

Instead of using Console_Input PAMSG you can use Send_Client_Text. It uses less resources on the server.

Subject: Re: [SSGM Plugin] InvalidChars
Posted by Neijwiert on Thu, 14 Aug 2014 12:18:44 GMT
View Forum Message <> Reply to Message

iRANian wrote on Thu, 14 August 2014 00:26Instead of using Console_Input PAMSG you can

use Send_Client_Text. It uses less resources on the server.

If you don't know how to do that:
Send_Client_Text(L"insert bullshit here", TEXT_MESSAGE_PRIVATE, true, -1, ClientReceiverID, true, true);

Not tested but it should look like that sort of.

Also when you use a certain function that iterates over a list ('Find_Player') then I advise you to store that result in a variable. Since you use it like a few times in a row on the same ID in the same function. It gets rid of unnecessary iterations.

Subject: Re: [SSGM Plugin] InvalidChars
Posted by roszek on Fri, 15 Aug 2014 04:20:51 GMT
View Forum Message <> Reply to Message

I decided to redo the plugin a bit. Check the original post.

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by iRANian on Fri, 15 Aug 2014 09:17:34 GMT
View Forum Message <> Reply to Message

In CheckName() instead of looping over every player to see if the name you want to set a player's name to is already used you can just use the function to get a cPlayer*, GameObject* or playerID by player name. If any of those functions don't return anything valid the name isn't in use.

There's also another thing you can speed up in the CheckName() function, isntead of doing:

```
if(stristr(StringClass::getFormattedString("%ls",name),bad_words[i]))
```

You should store the WideStringClass of 'name' before entering the loop, as you're executing getFormattedString to create a StringClass from the WideStringClass everytime when entering the loop.

While reading in the bad words list instead of using WordCount= to find out how many bad words there are in the list you can use the following two INIClass functions:

```
int Entry_Count(char const *section) const;
const char *Get_Entry(char const *section,int index) const;
```

Is this function call in OnLoadGlobalINISettings() needed?:

---

word_list.Delete_All();

If you want to make the 'check chars' functionality quicker you can create a bool array of 256, and setting the value of the array to 0 ('false') in the global INI, using memcpy(). Then read in the InvalidChars string from the INI and loop over every character of the InvalidChars string, convert each character to int and use this int as index in the bool array, to set the index to 'true'. Then in the CheckChars() function simply loop over every character of the player name, convert the character to int and use the int as index into the global bool array, if the bool at that index is set to 'true' then the player name contains an invalid char.

Something like this:

```
bool CharacterInvalidArray[256];

void CheckPlayerName::OnLoadGlobalINISettings(INIClass *SSGMIni)
{
 memcpy((char*)&CharacterInvalidArray, 0x0, 256 * sizeof(bool))

 StringClass InvalidCharsString;
 SSGMIni->Get_String(InvalidCharsString, "InvalidChars", "Invalid", "");

 for (int i = 0; i < InvalidCharsString.Get_Length(); i++)
 {
  int Index = (int)InvalidCharsString[i];
  CharacterInvalidArray[Index] = true;
 }
}

bool checkChars(WideStringClass name, int ID)
{
 StringClass SimpleName = StringClass(name);
 bool change_name = false;
 for (int i = 0; i < Simplename.Get_Length(); i++)
 {
  int Index = (int)SimpleName[i];
  if ( CharacterInvalidArray[Index ]== true )
  {
   SimpleName[i] = '_';
   change_name = true;
  }
 }

 if (change_name)
 {
  Find_Player(ID)->Set_Name(SimpleName);
 }
```

```
   return(change_name);

}
```

Note that you can justserveers with Chinese characters (hence why PlayerName is WideStringClass instead of just StringClass) so you should also check if the playername only contains ANSI characters when joining.

You can get the WideStringClass player name on join and use WideStringClass::Is_ANSI() to check if there are non-ANSI/ASCII characters. The definition for Is_ANSI() is

```
bool WideStringClass::Is_ANSI()
{
 if (m_Buffer)
 {
  for (int i = 0;m_Buffer[i] != 0;i++)
  {
   unsigned short value = m_Buffer[i];
   if (value > 255)
   {
    return false;
   }
  }
 }
 return true;
}
```

You can create your own function which will replace non-ANSI characters, variable 'value' > 255 in the above function, with underscores or something.

What Dragonade does (in the connection request event)is:

```
 for (int i = 0;i < Request.clientName.Get_Length();i++) {
  if (Request.clientName[i] < 33 || Request.clientName[i] > 126) {
   Request.clientName.RemoveSubstring(i,1);
   i--;
  }
```

Where Request.clientName is WideStringClass.

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by roszek on Fri, 15 Aug 2014 11:43:58 GMT
View Forum Message <> Reply to Message

That is some good advice.

Didn't think of doing playerID by player name instead of a loop.

also cleaned it up a bit based on what you posted.

The non-ansi chars I might look into a bit later.

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by Ethenal on Fri, 15 Aug 2014 13:52:35 GMT
View Forum Message <> Reply to Message

Now keep in mind that iran gives great programming advice, however the more intensive "speed tricks" aren't going to give you the slightest noticeable performance on current hardware with a game written 12 years ago - you gotta balance your extra work vs. performance

P.S. but that's not an excuse to leak memory!

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by iRANian on Fri, 15 Aug 2014 15:11:50 GMT
View Forum Message <> Reply to Message

They're useful tips for future use, if you apply them now you'll internalize thinking that way. The use of a boolean array for every possible ASCII value (i.e. array of 256 bools) to check if a character is invalid also looks simpler and cleaner.

Subject: Re: [SSGM Plugin] InvalidChars
Posted by Jerad2142 on Fri, 15 Aug 2014 19:52:13 GMT
View Forum Message <> Reply to Message

iRANian wrote on Thu, 14 August 2014 01:26Instead of using Console_Input PAMSG you can use Send_Client_Text. It uses less resources on the server.
You must have some really shitty servers if your concerned about something that happens ~50 times in a half hour period (less than ~50 because it would only happen on bad player names, and in all honesty I sort of feel like ~50 joins in 30 minutes still seems like a lot).

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by Xpert on Sat, 16 Aug 2014 03:30:15 GMT
View Forum Message <> Reply to Message

It's not a matter of the server being shitty. It's just an alternate method that uses less resource over time.

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by Ethenal on Sat, 16 Aug 2014 13:23:17 GMT
View Forum Message <> Reply to Message

iRANian wrote on Fri, 15 August 2014 10:11They're useful tips for future use, if you apply them now you'll internalize thinking that way. The use of a boolean array for every possible ASCII value (i.e. array of 256 bools) to check if a character is invalid also looks simpler and cleaner.
I agree with that, but the Send_Client_Text thing is absolutely pointless in my opinion. You could just make a wrapper function to take care of all of that, but like I said, I don't see why you should use a 5 argument function to do something that can be done much more easily for "performance" reasons.

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by iRANian on Sat, 16 Aug 2014 15:30:19 GMT
View Forum Message <> Reply to Message

Using console input is a lot slower and you can even notice a small FPS drop when it's used.

Subject: Re: [SSGM Plugin] CheckPlayerName
Posted by Jerad2142 on Sat, 16 Aug 2014 19:33:21 GMT
View Forum Message <> Reply to Message

iRANian wrote on Sat, 16 August 2014 09:30Using console input is a lot slower and you can even notice a small FPS drop when it's used.
Definitely depends on how old the hardware is then; because I was using it to debug brake lights in 4.2, so it was firing off a couple thousand Console_Inputs second and that was with the string being stuck together with sprint each time.