

---

Subject: TSR [TeamSpeak Regulator] FAQ/Info  
Posted by [Blazer](#) on Fri, 30 May 2003 09:56:13 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Lots of people are asking what this "TSR" is that I am working on. Similar to BR, I had an idea and vision of the way "things should be" in regards to integrating TeamSpeak with Renegade, but just like Renegade, the server did not support the commands or procedures necessary to make it happen.

Fortunately, the TeamSpeak developers are wonderful folks who actually listen to feature requests. I submitted requests to add a feature to allow the server to force teamspeak players into a password-protected channel, and commands for listing players and what channels they were in. They have currently added the features I asked for in the latest development build, and TSR is now possible!

What is TSR?

TSR is a server-side application that monitors both the Renegade FDS logfile and TeamSpeak server logfile. TSR has code which allows it to interface with the Teamspeak server (and FDS) to execute commands on said servers. TSR also has an option to place a "Bot" on IRC for administrative monitoring and control.

What benefit is TSR to players? What does it do?

TSR's main function is to automatically sort players logged into the Teamspeak server into their respective GDI and Nod voice channels. If a player switches teams, or is remixed at the end of a map, TSR will automatically move them into the proper channel. TSR will optionally broadcast to the FDS (on command or timed interval) the number of/which players using voice communication.

How does it work?

TSR, using the same code and method as BR, keeps in memory the Renegade FDS player names, IPs, & teams. It also keeps a list of the names, IPs, & channels of the people logged into teamspeak. Anytime the following happens:

1. Player joins Renegade
2. Player joins Teamspeak
3. Team Switch
4. Team Remix

TSR compares the TS login data with the FDS login data. It then places them into the proper channel (GDI or Nod).

What about people pretending to be someone else, like registering the TS name "Tankcmdr6" etc?

To thwart this the TS server is setup to ONLY ALLOW REGISTERED LOGINS. This means if you do not already have an account, you cannot connect to the Teamspeak server. To create an account players will page the FDS host with "tspass <password>". TSR will then create an account for them using their WOL name and password specified. This will ensure that everyones TS name matches their WOL name, and nobody can masquerade as someone else. This is also required to handle situations where multiple people play/login from the same IP (like Crimson and I) ... with the IPs being the same, the only way to tell "who is who" is if their TS name matches their WOL name.

Won't TeamSpeak slow down my server or negatively impact bandwidth?

TeamSpeak uses ZERO bandwidth when nobody is talking. Even with the high quality codecs, it uses very little. You of course have the option to use the lower quality codecs like CELP 5.1 which are but a drop in the bucket even at full utilization.

I don't want a bunch of people connected to my server chit-chatting about who knows what. The default setup for TSR is for the server to be game-only. In other words, if you are not in the game, you cannot be in the server. I will have other options such as a grace period, where when people join they have 1-3 mins to login to the FDS or be booted. This will ensure the server is used for game-chat only. If you WANT to have a social channel that is except from the nazi rules , I can add options for that too, let me know.

What the heck is "PERL" anyway, why not make it a cool GUI prog?

Being a "Unix Guy", PERL is my native language I have zero experience coding in Visual Basic or Visual C++, which is why I enlisted Dantes help in the creation of BR. I know how to program and can flowchart pseudocode, but I do not have knowledge of the syntax or experience to make a VB application. PERL is powerful in dealing with any kind of text manipulation, which is exactly the bulk of BR/TSR's processing. I'm making every effort to make TSR work under windows, in fact I am developing it in a win32 environment, and so far its working great, using only 4M of memory and no discernable CPU usage. Beta testing on larger servers will provide more data for realworld statistics.

What about abusive people (playing music, static, swearing etc)?

TSR can have moderators, who via IRC or in-game (/page server command) will have the power to devoice, kick, or ban people depending on their moderator access level. Although voice communication software has been around for quite some time, apparently many people are not accustomed to using it, and often mistakenly use "voice activated" mode (always use PUSH-TO-TALK mode!), with the threshold set too sensitive, and you get a lot of "darth vaders". I am going to submit another feature request to enforce push-to-talk mode, but I don't know if they will be receptive to it or not.

Any other questions, just post them here.

-Blazer

---