StealthEye wrote on Wed, 10 September 2014 14:39That would break the function for all servers not having special packages, so that's not making things any better. Adding a configuration option only makes things more complex. Why does it crash the clients though? Afaik, it shouldn't. That should be fixed instead.

You can override any default generated game definitions by just using the same name in a game definition in tt.cfg. If that's not enough to solve the issue, the bot should filter which maps should be available for change map commands instead of the engine, the engine should simply list what is supported, which is what it does now. If anything, it could show an extra entry for each listed game definition, stating whether it's user defined or generated.

At least put the listgamedefs console function in the public source code so we can change it. Its not very difficult to use a regular expression to filter things out. I mean this isn't exactly a complicated function. It doesn't have anything to do with netcode etc. Its a pretty simple method that traverses through a data structure and prints out each entry on a line.

The reason I would prefer it to be done through the FDS rather than the bot is if you have a lot of packages, the printing of the list takes twice as long and twice as many lines than if it were to only print the ones with explicit game definitions in the tt.cfg. This is why brenbot's setnextmap doesn't work until the entire list has been printed.

Adding a global package list would allow the client crashing issue to be fixed. The issue is if you have a global server objects.ddb with custom presets, and someone sets a map without typing out the entire definition name character for character brenbot will set it to just the mix file instead of the gamedefinition. Because of this it doesn't load the custom objects.ddb that is in our auxiliary package. Right now, there is no global package setting in the tt.cfg. If one was added it would be ideal that it would have two different entries. One entry that is loaded prior to the map specific packages and one entry that is loaded after. For example PreMapGlobalPackages and PostMapGlobalPackages. That way the priority of the global packages can be configured as the server operator would like.