

Game Server Hosting Reference (Multiplayer PC Games)

7 Days to Die

- **Config File:** Uses a `serverconfig.xml` file for server settings (world, difficulty, player limits, etc.). This XML defines all game parameters (e.g. day length, loot respawn) that the server loads on startup ¹. Edit it with a text editor and save changes before restarting the server.
- **Default Ports:** UDP 26900 is the default game port ². The server also uses a Telnet control port (usually 8081 by default, for remote console access) and a web map port if enabled. Ensure 26900 (UDP) is open/forwarded for players to connect.
- **Logs & Files:** The server's runtime log is written by the Unity engine. On Windows dedicated server, check the `7DaysToDieServer_Data/output_log.txt` (or timestamped `output_log__<date>.txt` if using the batch script) ³ ⁴. Crash dumps (`crash.dmp`, `error.log`) are in the main game folder under a dated directory ⁵. These logs contain errors (like block or AI exceptions) useful for troubleshooting.
- **Mod Support:** Supports server mods (e.g. overhaul mods like Darkness Falls). Mods are installed by placing mod folders into the `Mods/` directory under the game root. The server will load any mods on startup – all clients must have the same mods. Certain settings deviations (like changing gameplay values in XML) can flag the server as “modded” in the server browser ⁶. Always check server logs for lines indicating “modded” status if your server isn't showing as vanilla.
- **Troubleshooting:** If the server isn't visible to others, ensure **Windows Firewall** or your host's firewall allows 7DTD on UDP 26900 (and Telnet port if needed). For connectivity issues, verify that **Steam networking** is enabled (`ServerVisibility` setting). Regularly back up your save game folder (Worlds folder) to protect against map corruption. If players see mismatched world errors, delete their local copy of the world (so it re-downloads) or wipe the server's `save` for a fresh world. Memory leaks or stutters after long uptime can be mitigated by periodic restarts (7DTD is known to benefit from daily restarts on busy servers).

ARK: Survival Evolved

- **Launch Parameters:** ARK's dedicated server supports numerous startup options passed on the command line (after the map name). For example: `-automanagedmods` (auto-download/update mods on start) ⁷, `-EnableIdlePlayerKick` (kick idle players after a period) ⁸, `-ForceAllowCaveFlyers` (allow flyers in caves) ⁹, etc. Most gameplay settings are specified as `?Option=Value` in the map URL (e.g. `TheIsland?SessionName=MyServer?MaxPlayers=20?AllowThirdPersonPlayer=True`). There are dozens of such options – see official config lists ¹⁰ for the full range. Options not given on the command line will use their default values.
- **Config Files:** Aside from launch params, persistent settings are in INI files. **GameUserSettings.ini** (under `ShooterGame/Saved/Config/<Platform>/`) holds server options under `[ServerSettings]` (e.g. PvE mode, taming rates) and active mod IDs. **Game.ini** holds advanced game configuration (spawn rates, engrams, etc.). These files are generated after first run and can be

edited to tweak server behavior ¹¹. Always stop the server before editing INIs, and ensure no syntax errors (ARK is sensitive to proper formatting).

- **Default Ports:** By default the server listens on UDP 7777 for game client traffic, with a second UDP port 7778 (game traffic +1) ¹². The Steam query port defaults to UDP 27015 ¹² (for the server browser listing). ARK's RCON (remote console) runs on TCP 27020 by default ¹² – this allows remote administration if enabled (`-rconenable`). Make sure to open/forward **7777/UDP** (game) and **27015/UDP** (query) at minimum, plus **27020/TCP** if using RCON.
- **Logs & Data:** ARK writes logs to files in the `ShooterGame/Saved/Logs/` directory ¹³ (log files have names like `ShooterGame.log` or `ServerLogs*.log`). Crashes may produce `.dmp` files in this folder as well. All world save data and player/tribe profiles reside under `ShooterGame/Saved/SavedArks` (for each map) or `SavedArksLocal` ¹⁴ – back this up regularly. If the server fails to save data (e.g. world resets), check file permissions on this folder and free disk space.
- **Mod Support:** ARK servers support Steam Workshop mods. To install mods, you can specify mod IDs in `GameUserSettings.ini` (`ActiveMods=<IDs>` comma-separated) or on the command line via `?GameModIds=`. The `-automanagedmods` flag automates downloading/updating those mods on server start ⁷ – recommended for keeping mods in sync. Ensure the server has the same mod versions as clients (mismatched mods cause connection timeouts). Total Conversion mods or custom maps require launching with the appropriate map name or TC ID. Mods are stored in `ShooterGame/Content/Mods` on the server. After adding new mods, always restart the server and allow it time to download mods before it becomes joinable.
- **Troubleshooting: Server not appearing in browser:** Double-check that the query port is accessible (use Steam's server list by IP to test). Sometimes ARK servers won't show in the in-game list due to Steam query issues – connect via IP:Port as a workaround. **Lost characters or progress:** This usually means the server didn't save properly. Always shut down using in-game save (or `Ctrl+C` in console, or `saveworld` admin command) to write data. For safety, periodically copy the `SavedArks` folder as backup ¹⁴. **High CPU, low RAM on Linux:** ARK on Linux can show high CPU usage but under 1GB RAM use – this is normal due to how memory is allocated (it doesn't necessarily indicate an issue) ¹⁵. **Mod issues:** If the server crashes or fails to start after a mod update, try removing the most recent mod or start with `-TotalConversionMod=0` to disable TC mods. Check the log in `Saved/Logs` for lines referencing mod load errors or conflicts. Finally, keep server and mods updated – outdated servers or mods can lead to version mismatch errors preventing clients from joining.

Arma 3 (and DayZ Standalone)

- **Config Files:** Arma 3 dedicated server is configured via a `server.cfg` (mission rotation, passwords, etc.) and optional `basic.cfg` (network tuning). Place these in the server's root or profile directory and launch the server with `-config=server.cfg -cfg=basic.cfg`. DayZ Standalone uses a similar `serverDZ.cfg` for config. The `-profiles=<folder>` launch parameter in Arma/DayZ determines where logs and config (like `*Users\<Name>*` files) are stored; using a custom profile folder is recommended to keep files organized.
- **Default Ports:** Arma series servers default to UDP port **2302** for game data. Additional consecutive ports are used for multiple instances or services: by default Arma 3 uses UDP **2302-2306** (inclusive) for one server ¹⁶. (2302 is game traffic; 2303/2304 for VoIP or admin, 2305/2306 for Steam query and VAC). Ensure UDP 2302-2306 are open. DayZ Standalone also defaults to UDP 2302 (with 27016 for Steam query). For RCON in Arma, if using BattlEye RCON, set a RCON port (default 2302 TCP) and password in `BEServer.cfg`.

- **Logs:** Arma servers generate `.rpt` log files containing extensive information (startup configs, scripts errors, etc.). On Windows, these are in `C:\Users\<User>\AppData\Local\Arma 3\` (or the `-profiles` folder). On Linux, the server by default outputs logs to stdout (console) instead of a file ¹⁷ ¹⁸, so you should redirect output to a file (e.g. `./arma3server ... >> server.log 2>&1`) to save logs. Always check the RPT log for issues like mission errors, mod load failures, or players being kicked (e.g. BattlEye reasons).
- **Mod Support:** Arma 3 supports mods and custom missions. Mods are typically server-side as well as client-side: if you run mods, players must have the same mods loaded. Use the `-mod=` launch parameter to load mods (specify semicolon-separated folder names or use `-mod=@mod1;@mod2`). For Steam Workshop mods, you can supply the directories or use the `-mod=@workshop**` syntax after copying from the !Workshop folder. The server doesn't auto-download mods – you must install them manually on the server. Use a tool like SteamCMD or a launcher to keep mods updated. DayZ server also supports mods (via `-mod=`) and has a `mods` folder for content; ensure to set `verifySignatures` appropriately if requiring client signatures.
- **Troubleshooting: Mission not loading or server stuck at waiting for players:** This can indicate a scripting error in the mission or a mod issue. Check the RPT log for errors (e.g., missing addons). **Players can't connect or are kicked:** Ensure **BattlEye** isn't kicking for mod signatures (or disable `verifySignatures` for testing). Also, if using a headless client, verify it's whitelisted. **Performance:** Arma servers are CPU-intensive; consider using `-enableHT` for hyper-threading utilization and tweak `serverMaxFPS` in `basic.cfg`. If experiencing desync, monitor bandwidth and set an appropriate `MaxPing` or player limit. For DayZ, persistence issues (loot not saving) often relate to the storage files – always shut down cleanly to save the world state.

Counter-Strike 1.6 / Half-Life (GoldSrc Engine)

- **Server Software:** Uses the Half-Life Dedicated Server (HLDS) for games like Half-Life Deathmatch, Counter-Strike 1.6, Day of Defeat, etc. HLDS is typically managed via SteamCMD (app 90) or the old `hldsupdate` tool. After installation, the main folder contains subfolders per mod (e.g. `cstrike/` for CS). Run `hlds.exe` (or `hlds_run` on Linux) with `-game <mod>` (e.g. `-game cstrike`) to host a specific game mod.
- **Config Files:** The primary config is `server.cfg` in the mod's folder (e.g. `cstrike/server.cfg`) ¹⁹. This is executed on server start and sets game CVARs (hostname, maxplayers, friendly fire, etc.). Other configs: `mapcycle.txt` for map rotations, and `motd.txt` for the server's MOTD. Admin mods like AMX Mod X or Metamod have their own configs (e.g. `addons/amxmodx/configs/`). Always set `sv_lan 0` in `server.cfg` to ensure the server advertises to the internet ²⁰ ²¹.
- **Default Ports:** HLDS uses UDP **27015** for game traffic by default (and this is the port clients connect to) ²¹. It also uses TCP **27015** for RCON (remote console) ²¹. The Half-Life TV (spectator proxy) uses UDP **27020** if enabled ²¹. Additionally, HLDS registers with the master server on UDP **26900** (this is automatic for server listing) ²². Make sure to open/forward 27015 UDP (and TCP for RCON). If hosting multiple HLDS instances on one machine, each needs unique ports (set via the `-port` launch parameter for game port and `+clientport` for the client-side port which defaults to 27005).
- **Logs:** By default, HLDS can log to both console and files. Enabling the `log on` setting will create log files under the mod folder (e.g. `cstrike/logs/Lxxxx.log`). These logs record player actions, chat, and some errors. For more detailed debugging, launch the server with `-dev` or use third-party logging plugins. RCON feedback and errors also appear in the console output (use a tool like Screen on Linux to keep the console running or redirect output to file).

- **Mod/Addons:** Popular admin mods include **Metamod** and **AMX Mod X**. To install, you typically add a Metamod DLL (or SO) to the `dlls/` and update `liblist.gam` to point to it, then add AMX Mod X as a Metamod plugin. These allow additional plugins (for admin commands, stats, etc.). Ensure the versions match your HLDS build (use latest AMX Mod X for Steam HLDS). For content mods (like custom maps or models), simply add them to the appropriate folders and update maps configuration (e.g. add to `mapcycle.txt`). Clients will download custom content from the server if `sv_downloadurl` is set (you can configure FastDL with an HTTP URL for speed).
- **Troubleshooting: Server not visible/public:** Ensure `sv_lan 0` and that you have a valid **Steam Game Server token** if required (older GoldSrc don't require tokens, that's more Source games). Also confirm your external IP isn't LAN (check `sv_ip` or launch with `-ip <publicIP>`). **High ping or choke:** Check the **rate** settings – for internet play, set `sv_maxrate` and `sv_minrate` appropriately (e.g. `sv_maxrate 25000`). Too low rate can choke players, too high can saturate. **RCON not working:** Remember RCON uses TCP 27015 – if you can't remote in, verify that port is open and that you set `rcon_password` in `server.cfg`. Note that repeated wrong RCON attempts will temporarily blacklist your IP. **Crashes:** Update to the latest build via SteamCMD (valve & beta branches for HLDS). Also disable any recently added plugins to pinpoint issues. HLDS is stable but custom addons can cause instability.

Source Engine Games (Counter-Strike: Source/GO, Team Fortress 2, Garry's Mod, Left 4 Dead, etc.)

- **Server Software:** Source games use SRCDS (Source Dedicated Server). Obtain via SteamCMD (e.g. app 232330 for CS:GO, or 4020 for Source DS base). Many Source games now use **Steam Game Tokens** for public servers – generate a token on Steam Game Server Account Management and supply it via `+sv_setsteamaccount <token>` in the launch parameters (especially for CS:GO, TF2, etc.). Launch with `srcds.exe -game <folder>` (like `-game csgo` or `tf`) and `+map <mapname>` to start.
- **Config Files:** Primary config is `cfg/server.cfg` in the game's directory (e.g. `csgo/cfg/server.cfg`). This sets cvars such as hostname, rcon_password, game mode settings, etc. Additional configs can be in `cfg/autoexec.cfg` (executes on map load) or specific ones like `gamemode_server.cfg` for CS:GO. For games with map-specific configs, you can use the `cfg/maps` folder. **Log on** is enabled by default for most Source games, and logs will appear in `logs/` directories.
- **Default Ports:** Most Source engine servers default to **UDP 27015** for game traffic (also used for queries) ²³. RCON also shares this port (but over TCP) ²³. SourceTV (for games that support it, e.g. CS:S/CS:GO spectator relay) uses **UDP 27020** by default ²⁴. Additionally, a Source server will use a small range around these: e.g. a client port (outgoing) on 27005 UDP ²⁵, and may utilize ports 26900+ for Steam backend. In summary, open UDP 27015 (game/query) and UDP 27020 (SourceTV) at minimum; if using RCON, open 27015 TCP as well ²³. For multiple servers, set `-port`, `+tv_port`, and `-clientport` to avoid conflicts (by default, a second server might try 27016, etc.).
- **Logs:** Source servers log to files in the `logs` folder under the mod dir (e.g. `csgo/logs/` or `orangebox/tf/logs/`). These logs (often named by date/map) record chat, kills, connections, and errors. They are useful for gameplay logs but not detailed engine debugging. For more detail, use `-condebug` to log console output to a file (`console.log`) or third-party tools (like mods that log admin actions). Crash dumps (e.g. `.mdmp` files) may appear in the root folder if the server crashes. Always

check the console or `console.log` for errors (like missing map, bad config syntax) if the server won't start.

- **Modding & Workshop:** Many Source games allow mods via server plugins (e.g. **MetaMod:Source** and **SourceMod** for admin tools and gameplay mods). These require adding a `metamod.vdf` to the addons folder and placing plugins accordingly. For example, to use SourceMod, install **Metamod:Source**, then **SourceMod**, then configure admin rights in `addons/sourcemod/configs/admins.cfg`. Content like maps or models can be added directly. **Steam Workshop integration:** Some games (CS:GO, Garry's Mod) support pulling maps or addons from Workshop. For CS:GO, you can host a collection of Workshop maps by using `+host_workshop_collection <ID>` and `+workshop_start_map <map>`. Garry's Mod servers commonly use Workshop collections for addons – configure the collection ID and set up an autoloader (in `garrysmo/cfg`). Clients will auto-download these Workshop items when joining.
- **Troubleshooting: Server not listed:** Ensure you've set the correct game mode (for CS:GO, `gamemode0x.cfg` execution) and that a map is running (servers don't show until a map is loaded²⁶). Also verify your Steam GSLT token is correct (server console will show an error if not). **RCON issues:** Use a strong **rcon_password** and remember to open the port. Use a tool like **rcon tool** or in-game console to test (e.g. `rcon_address` and `rcon_password` from a client). **Performance:** Source servers are single-threaded; high player count or tickrate (e.g. 128-tick CS:GO) will strain CPU. Lower the tickrate (with `-tickrate 64` or 32) for less CPU usage, or upgrade hardware for better performance. **Crashes:** Update to the latest server version (SteamCMD) and ensure mods (SourceMod plugins) are up-to-date. Try running without any addons (rename the addons folder temporarily) to see if stability improves. Some Source games like GMod can crash from certain Lua addons – check the `LuaError` logs in GMod for clues. Always keep backups of config files and a vanilla copy of the server to test with.

Minecraft (Java Edition)

- **Server Software:** Official server .jar (from Minecraft.net) or third-party distributions (Paper, Spigot, etc. which offer more features and optimizations). To start, run `java -Xms1G -Xmx4G -jar minecraft_server.jar nogui` (adjust RAM as needed). Accept the EULA by editing `eula.txt` (set `eula=true`) on first run. For modded Minecraft, use the appropriate server jar (Forge, Fabric, etc.) and ensure clients use the same modloader and mods.
- **Config Files:** The primary config is `server.properties` in the server folder. This contains settings like level name, difficulty, PVP on/off, whitelist enforcement, etc. It's a plain text file – after editing, restart the server for changes to apply. Notable entries: `max-players`, `online-mode` (auth), `level-seed`, and network settings like `server-port` and `server-ip`. By default **server-port=25565**²⁷, which is the main game port. There are also operator and whitelist files (`ops.json`, `whitelist.json`) and **banned-players.json** for bans. If using Spigot/Paper, additional config files (`paper.yml`, `spigot.yml`, `bukkit.yml`) will appear for tweaking those servers.
- **Default Ports:** **25565 TCP** is the default port for Minecraft multiplayer²⁷. The server uses TCP for game data (clients initiate a TCP connection). UDP is not used by the vanilla server (some plugins or proxies might). If running multiple servers, each needs its own port (e.g. 25566, 25567, etc.). Port-forward 25565 (TCP) on your router for internet play. For Bedrock Edition (if relevant), default ports are 19132 UDP, but this guide focuses on PC Java Edition.
- **Logs & Crash Reports:** Minecraft server logs everything to the `logs/` folder. The latest log is `logs/latest.log` (text file), and on each restart, the previous log is archived (gzip). Check `latest.log` for server startup progress, player join/leave, errors (stack traces from mods/plugins). The

console output goes here as well. In case of a severe crash, a **crash report** will be generated in the `crash-reports/` folder, containing a stack trace and state info at time of crash. This is extremely useful for diagnosing issues like mod conflicts or out-of-memory errors. Always allocate enough RAM (via `-Xmx`) to avoid **OutOfMemory** crashes (which would be noted in logs).

- **Modding & Plugins:** Vanilla server has no mod/plugin support. To use mods, run a Forge or Fabric server (matching the client modloader). Mods are then added to the server's `mods/` folder just like the client's. **Plugins** (which don't require client mods) are supported via server forks like Spigot/Paper (which implement the Bukkit API). Place plugin jars in the `plugins/` folder. Common plugins include EssentialsX, WorldEdit, and permission systems. For large modpacks, use the official server pack if provided, or ensure all mods and config files are identical between client and server. **Steam Workshop:** N/A for Minecraft, but mods are distributed via community sites (CurseForge, Modrinth). Some server managers use modpack IDs to auto-download packs. Always verify mod compatibility with your server version (a mod built for 1.16 will not work on 1.20, etc.).
- **Troubleshooting: Can't connect (Timeout):** Check firewall and that you're using the correct IP and port. If the server says "Waiting for connection" and never shows in the client, ensure `server.properties` `server-ip` is either blank or correctly set (usually leave it blank so it binds to all interfaces). **"Outdated server" or "Outdated client" errors:** The server and client versions don't match – update one side accordingly. **Lag and tick delays:** Monitor the server tick rate in logs (Spigot's `/tps` command). If TPS drops below 20 consistently, you may need to reduce loaded chunks (lower render distance in `server.properties`), limit entity counts (via gamerules or plugins like ClearLagg), or upgrade hardware. **Crashes on startup:** The crash report or latest.log will usually indicate the mod or plugin causing issues. Remove or update the offending mod. For vanilla, remove any datapacks that could be crashing the server (check the log for datapack errors). Also ensure you have the correct Java version – newer Minecraft (1.17+) requires Java 17; using the wrong Java can prevent launch (the log will say if so).

Quake III Engine Games (Quake 3 Arena, RTCW: Enemy Territory, Call of Duty 1/2)

- **Server Software:** Many late-90s/early-2000s shooters use the id Tech 3 engine or derivatives. **Quake III Arena** and its mods (e.g. Urban Terror) run on `ioq3` or the original dedicated binary (e.g. `quake3.exe +set dedicated 2`). **Return to Castle Wolfenstein** and **Wolfenstein: Enemy Territory** have their own server executables (`wolfded.exe`, `etded.exe`). **Call of Duty (1/2)** also inherited a similar console-based server style. You typically start these via command line with parameters or a config. For example, Quake3: `quake3 +set dedicated 2 +exec server.cfg +map q3dm1`. Enemy Territory: `etded +set dedicated 2 +exec server.cfg +map oasis`.
- **Config Files:** All these games use a text config (filled with CVARs) to set up the server. For Quake3/ET, the default server config is often named `server.cfg` placed in the game's base folder (`baseq3/` for Quake3, `etmain/` for ET). You set variables like `sv_hostname`, `sv_maxclients`, `timelimit`, `rconPassword`, etc. Mods will have their own folder (e.g. `etpro/` for ET Pro mod) with separate configs. It's common to create multiple configs (like one for each gametype or match settings) and exec them as needed. After editing configs, either exec them via console or restart the server.
- **Default Ports:** **27960 UDP** is the default port for Quake 3 Arena and games based on its engine ²⁸. Enemy Territory also uses 27960 UDP by default (and will increment for additional servers on same machine, e.g. 27961). Call of Duty (which branched from id Tech 3) typically uses **28960 UDP** as default. These servers communicate over UDP only. Ensure UDP 27960 (or 28960 for CoD) is open.

The server will also use the next port for queries by master server and possibly for PB (PunkBuster) if enabled. For multiple instances, you can set **net_port** cvar to a different number.

- **Logs:** Quake engine servers can log to a file by setting `g_log` (for game event logs) and `g_logsync 1` (to flush each write). By default, if not set, they may only print to console. It's wise to enable logging for moderation (chat, kills, etc.). Logs will be created in the mod folder (e.g., `etmain/logs/`). There is also a *console* log (set by `+set logfile 1`) which captures the server console output to `qconsole.log`. This is useful for debugging crashes or startup issues (it will show any config errors or missing files).
- **Mod Support:** Quake 3 and ET are highly moddable. Running a mod usually means installing it in a folder and launching the server with `+set fs_game <modfolder>`. For example, to run the popular ET Pro mod, you'd use `+set fs_game etpro` and have the mod files in an `etpro` directory. Quake 3 mods (like CPMA, Urban Terror for Q3) similarly go in their folder. Mods can introduce new CVARs – read their documentation for specific server settings. Ensure clients have the mod as well (for pure servers, they will auto-download small files, but large mods often are distributed separately). For Call of Duty, mods are often custom gametype scripts; you load them via config or command-line as well.
- **Troubleshooting: Server not visible:** Make sure *dedicated 2* is set (this tells the engine it's an internet server). Also check `sv_allowDownload` and `sv_punkbuster` if applicable – misconfigurations there won't hide the server but can affect connectivity. Quake engine games rely on master server heartbeat – by default they send heartbeats to id's master. If a server doesn't show up, the master server might be down or your server can't reach it (ensure outbound connectivity on UDP 27950–27960 range). **RCON:** Set a strong `rconPassword`. You can use in-game console `\rcon address:port password command` to administer. If RCON isn't responding, you might have the wrong port or the server may be behind NAT without port forwarded. **Common crashes:** Often caused by mismatched mod versions or corrupted pk3 files. If clients get kicked for "Invalid pk3 files", ensure your custom maps or mods are correctly installed and not conflicting. Clean out any unnecessary .pk3 files from the server's base folder to avoid pure-server issues. For Enemy Territory on modern OS, run as admin or in compatibility mode if you see weird behavior (it's an older game). The community has patched binaries for ET (etlegacy) and Quake 3 (ioquake3) which can improve stability on new systems.

Rust

- **Server Software:** Rust's dedicated server is shipped via SteamCMD (app 258550). Run `RustDedicated.exe` with parameters or use a batch script. Example launch:
`RustDedicated.exe -batchmode +server.port 28015 +server.level "Procedural Map" +server.worldsize 3500 +server.maxplayers 50 +server.hostname "MyServer" etc.`
Rust is resource-intensive, so ensure adequate RAM and a good CPU for physics. After first run, default config files like **server.cfg** are not created automatically – you must create them (in `<installdir>/server/<identity>/cfg/`). The `<identity>` is a folder name you choose (via `-server.identity` param) to separate saves/configs for multiple servers on one machine ²⁹.
- **Config & Commands:** You can either use startup command-line variables (prefixed with `+server.` or `+rcon.`) or put them in a `server.cfg` file in the `cfg` folder ²⁹. The `server.cfg` is executed on server start, and you don't include the `+` or `-` prefixes inside it. Common settings: **server.port** (game port, default 28015) ³⁰, **server.ip** (if binding to a specific IP), **server.maxplayers**, **server.seed** (random world seed), **server.worldsize**, **server.saveinterval** (how often to save world, default 300s), **rcon.port** (default 28016) and **rcon.password** (set this!). Some settings cannot be changed via config

and require command-line (like level or world size). Rust also has an in-game console (or RCON console) for admin commands (e.g. giving items, banning players).

- **Default Ports:** By default, Rust uses UDP **28015** for game traffic ³¹ ³⁰. The RCON tool (e.g. WebRCON or RustAdmin) communicates on **28016** (TCP for RCON) ³¹. Additionally, Rust has a query port used for the server list – by default Rust now uses **UDP 27015** for query ³¹ (it can be changed via `+server.queryport`), and if not set explicitly it may default to game port +1 or Steam port ranges). Open/forward **28015 UDP** (game), **28016 TCP/UDP** (RCON) ³¹, and **27015 UDP** (query). If you run multiple Rust servers, each needs unique ports (e.g. 28015/28016/27015 for one, then 28017/28018/27017 for another, etc.).
- **Logs:** Rust servers do not log gameplay events to file by default. You can enable logging by setting `-logfile` parameter or using the `LOGFILE` option in some server panels ³². With a `-logfile "output.log"` startup param, the server will write console output to that file (including chat, join/leave, errors). Otherwise, you rely on the live console or an RCON tool to view events. It's highly recommended to enable a logfile for moderation purposes. Also, Rust saves user persisting data (blueprints, bans, etc.) in the `server/<identity>/` folder (e.g. `UserPersistence.db`, `player.deaths.json` etc.). Back up this entire folder regularly (especially before wipes or updates).
- **Modding:** Rust can be modded with **Oxide/uMod** (a third-party mod framework). Installing Oxide (now uMod) involves patching the server assembly, which enables use of plugins written in C# (extensions with `.cs` files). These plugins (like kits, economics, teleport, etc.) go into the `oxide/plugins/` directory. Always use the Oxide version matching the Rust server version (Oxide updates lag behind official patches by a bit). Clients do not need any mods – Oxide plugins run server-side. Be mindful that using mods will mark your server as “modded” in the server list (separate from the community list).
- **Troubleshooting: Server not on list:** If your Rust server doesn't appear in the server list, try direct connect (`client.connect IP:port` in Rust console). It may be due to the query port issue – ensure both the game port and query port are open and not the same number ³³. Also check that your server has a unique name and not “Untitled” (servers with default names might be hidden).
RCON not working: Make sure `rcon.password` is set and you're using the correct RCON port (default 28016). Some RCON tools require both TCP and UDP on that port ³¹ – open both to be safe.
Performance: Rust servers will automatically save regularly (default 5 min); if you see “Saving complete” taking too long, consider using an SSD. Entity count can grow large – you might need to wipe (clear save) every so often to keep performance optimal. **Crashes or memory leaks:** Rust is updated frequently; always update to latest version. A common admin mistake is running **too many entities** (huge player bases, etc.) – use plugins like EntityCleanUp or adjust decay settings if needed. If the server won't start after an update, try validating files via SteamCMD (or remove Oxide and wait for an update if you use mods). Always consult the `output_log` (if enabled) for exceptions or errors on startup.

Terraria

- **Server Software:** Terraria's dedicated server is included with the game (Windows: `TerrariaServer.exe`; for Linux, use the dedicated Linux server or mono). Launching it interactively will prompt for world selection and configs, but you can also use a config file or command-line flags. For example: `TerrariaServer -world "WorldName.wld" -port 7777 -maxplayers 8 -pass "mypassword"`. There's also a `serverconfig.txt` file (generated by the

server) which can hold settings so you can run the server non-interactively (`-config serverconfig.txt`).

- **Config Options:** Key settings include world (path or name), `worldsize` (if generating new), `maxplayers`, `port`, `password` (server password if any), and `motd`. These can be set in the `serverconfig.txt` (format is `setting=value` per line). For example: `port=7777` (default), `maxplayers=16`, `worldpath=/path/to/World.wld`, etc. You can also specify auto-create parameters to generate a new world if none exists (like `autocreate=3` for large world). Once configured, run the server and it will load or create the world and start accepting connections.
- **Default Port: 7777 TCP** is the default Terraria server port. Terraria uses TCP for client connections. If you change it, clients need to specify the new port when connecting. Make sure to port forward/open 7777 TCP. If running multiple instances, use different ports (e.g. 7778, 7779, etc.).
- **Logs:** The Terraria server console output can be directed to a file by launching via command line. By default, it prints to the terminal only. If using a host or script, you can pipe output to a log file for record-keeping. There isn't a built-in extensive logging system for Terraria vanilla server; however, some server mods (like TShock) provide logging of commands and events. If the server crashes, it may not produce a crash log, but you can sometimes see error info in the console (e.g. exception traces).
- **Modding/Tools:** Vanilla Terraria servers are fairly basic. **TShock** is a popular modded server that adds admin commands, region protection, and plugin support. Using TShock involves downloading its server version (which replaces `TerrariaServer.exe`) – it has its own config and uses SQLite for data. For purely vanilla but with ease of management, community tools like **Terraria Server Wrapper** exist (but not as common now). Note that if you use mods like `tModLoader`, the server needs to run the `tModLoader` server and all clients need those same mods – effectively turning Terraria into a different game version. The default server won't load content mods.
- **Troubleshooting: Connection issues:** Since Terraria is TCP, if players get “Connecting” timeouts, it's usually firewall or port forwarding not correct. Double-check the firewall (Windows often blocks Terraria server on first run – allow it on public networks). Also ensure the **GUID** (if using TShock) or any IP whitelisting isn't preventing connections. **Server performance:** Terraria is not very CPU heavy, but if you experience slowness, it could be large worlds or many NPCs – there are not many server-side performance tweaks except reducing max players or using a smaller world. **World file issues:** Occasionally a world might become corrupted (server crashes on load). Terraria auto-saves `.bak` backups of worlds – you can restore the `.bak` (rename it to `.wld`). Keep manual backups of your world after major builds or periodically. **Admin tasks:** Without TShock, giving items or teleporting players isn't possible on vanilla server (those require in-game inventory editors or mods). If you need those capabilities, consider running TShock which allows item spawning and more via commands.

Unreal Engine Games (Unreal Tournament series, Killing Floor 2, etc.)

- **Server Software:** The Unreal Engine 1/2 games (Unreal Tournament 99/2004) and Unreal Engine 3 games (e.g. Killing Floor 2) use dedicated server executables or commandlets. **Unreal Tournament 99 (UT99)** server can be started via `ucc.exe` (Unreal console command utility) or launching the game with `-server` flag. For example: `ucc server DM-Deimos? game=Botpack.DeathMatchPlus ini=UnrealTournament.ini log=Server.log`. UT2004 uses a similar `ucc.exe` in the System folder. **Killing Floor 2 (UE3)** has a dedicated server tool (available on SteamCMD, app 232130). Launch with `KFGame.exe server <map>?game=<GameMode> -log`.

- **Config Files:** Unreal games store settings in `.ini` files. UT99/UT2004 use `UnrealTournament.ini` (or `UT2004.ini`) for server settings (under sections like `[Engine.GameReplicationInfo]` for server name, etc., and gametype-specific sections). Killing Floor 2 uses `KFGame.ini` and `KFEngine.ini` (found in `KFGame/Config/` folder) for server config – you set things like MaxPlayers, Admin password, WebAdmin enabled, etc. For KF2, enabling Web Admin (a web-based control panel) is done in `KFWeb.ini`. Many parameters can also be passed on the command line when launching (map, gamemode, mutators, etc.). After first run, edit the `.ini` files to configure your server (e.g. KF2's `ServerName` in `KFGame.ini`).
- **Default Ports:** **7777 UDP** is a common default game port for Unreal/UE3 servers (e.g. UT99/2004 default to 7777 UDP; Killing Floor 2 also uses 7777 UDP for gameplay) ³⁴. **Query ports:** UT uses 7778+ for query and 27900 for master server heartbeat. Killing Floor 2's query (Steam query) is on **27015 UDP** by default ³⁴, and it also uses **20560 UDP** (Steam master port) ³⁴. KF2's Web Admin runs on **8080 TCP** by default ³⁴. So for KF2 open 7777/UDP, 27015/UDP, 20560/UDP, and 8080/TCP. For UT99, open 7777–7780 UDP and 27900 UDP (the community recommends 7777-7781 and 27900) ³⁵ ³⁶. UT2004 similarly uses 7777 UDP (plus 7778, 7787 for query, and 28902 for master). Always consult specific documentation, but generally these games primarily use UDP for game traffic and queries.
- **Logs:** UT and other Unreal servers log info to the console and a log file. In UT99, the `server.log` (as specified by `log=...` on startup) will contain all server output. This includes player join messages, map change, and any warnings (e.g. missing packages). Make sure to enable logging (most default shortcuts already do, but if launching manually include the `log=` parameter). Killing Floor 2 writes logs under `KFGame/Logs/` with names like `Launch.log` by default. These logs are crucial for diagnosing issues (like why a server didn't start a map, or crashes). If crashes occur, UT can produce crash logs (`UnrealTournament.log` with a crash callstack) or Windows `.dmp` files.
- **Mods & Mutators:** Unreal games have *mutators* and mods. A mutator is a small gameplay script (e.g. instagib, low gravity) that can be activated via command line or admin console (for UT, add `? mutator=Package.MutClass` to the startup line; for UT2004, similar). Full conversion mods might come with separate executables or require `-mod` switch. Killing Floor 2 allows mods via Steam Workshop – you can subscribe to maps or mods on the workshop and use their IDs in the server's config (`KFGame.ini`) under `[OnlineSubsystemSteamworks.KFServerWorkshop]` by listing Workshop IDs. The server will download those mods on startup. Ensure to also add mutators in the startup command if required for the mod. For example, custom maps in KF2 need to be added to the map cycle in `KFGame.ini` or via Web Admin.
- **Troubleshooting:** **Server not showing up:** In UT, make sure `ServerBehindNAT` is set correctly (UT2004 has this option) if you're behind NAT, and that the master server is not down (Gamespy master was used historically – now community master servers exist; ensure you have patches that update the master server address). **Killing Floor 2 not visible:** Ensure your Steam ports are open (27015, 20560) – KF2 uses Steam's master server and won't list if those are blocked. Also check that you set a unique server name; sometimes servers with default name might be filtered out. **Web Admin not working (KF2):** Make sure you enabled `bEnabled=true` in `KFWeb.ini` and perhaps added an admin password. Then connect via `http://<serverIP>:8080` (or whatever port set in `KFWeb.ini`). **Crashes:** If an Unreal-based server crashes on startup, often it's a missing package (`.u` or `.utx` file). The log will tell you which file is missing – install the required files (for example, a custom map's dependencies). Also, mismatched versions between server and client (especially for KF2 after a big update) can cause issues – always update your server when a game update hits. For UT, using the latest community patch (UT99 v451 or v469) is advised for stability on modern OS. **Performance tweaks:** UT servers can handle many players on modern hardware (just ensure tickrate isn't set

unreasonably high). KF2 servers benefit from turning off unused features (like web admin if not needed, or lowering `NetServerMaxTickRate` to limit FPS). Finally, always run these servers on wired connections with low latency; older engines aren't tolerant of high jitter.

Valheim

- **Server Software:** Valheim uses a dedicated server tool (available on Steam as “Valheim Dedicated Server” or via SteamCMD). It's essentially the game running in dedicated mode. Launch `valheim_server.exe` with a startup script that provides world name, server name, password, and ports. For example, the bundled Windows `.bat` includes: `valheim_server -nographics -batchmode -name "MyServer" -port 2456 -world "WorldName" -password "YourPass"`. Always run with `-nographics -batchmode` for a headless server.
 - **Config & Save Data:** Valheim doesn't have a traditional config file for its server; settings are passed via command line. Key parameters: **-name** (server display name), **-world** (save name; if non-existent, a new world will be created with that name), **-password** (required, cannot be blank due to game design), **-public** (1 or 0, to list on server list or not). The save files (world `.db` and `.fwl`) are stored in the `%AppData%\Valheim\worlds` (Windows) or `~/.config/unity3d/IronGate/Valheim/worlds` (Linux) by default, unless you use `-savedir` to specify a custom directory. Back up your world `.db` file periodically to protect progress.
 - **Default Ports:** By default, Valheim uses UDP **2456** for the main game port ³⁷. It also uses the next two ports (2457, 2458 UDP) for cross-play and query purposes. The official docs mention a range: “The default Port Range that the Server uses is 2456-2458” ³⁷. In practice, 2456 is the one clients connect to; 2457 is used to communicate with Steam for server listing, and 2458 may be used for extra services (some setups see activity on 2458 as well). **Open/forward UDP 2456-2458** to be safe. If you change the port via `-port` argument, remember it still consumes two subsequent ports as well. Clients only need to specify the first port when connecting.
 - **Logs:** The Valheim server prints output to console. The dedicated server `.bat` by default writes to a file `LogOutput.log` (this is configured in the `.bat` by using `-logfile`). Ensure your startup includes a `-logfile` parameter so you have a record. The log will show each player connecting (with their PlayFab ID), disconnects, world save messages (“World saved (X ms)”), etc. If the server crashes or exits, check this log for any exception or error message (e.g. often a mod mismatch will not crash the server but simply not allow connection, logging a relevant message).
 - **Modding:** Valheim can be modded using frameworks like **BepInEx** or **Valheim Plus**. To use mods, you must mod the server the same way as a client – e.g., install BepInEx on the server and add the same mod DLLs. All connecting players need the same mods for compatibility (Valheim will not let clients join if their code hash differs when using mods). Valheim Plus, for instance, allows changing gameplay settings; it uses a config file (shared between server and clients) to enforce consistent settings. When running a modded server, **make sure to announce required mods**, as the game's server browser doesn't indicate mod requirements. There's no official “mod support” or Workshop (mods are from Nexus or Thunderstore).
 - **Troubleshooting: Server not showing up or friends can't find it:** If using `-public 1` and it doesn't list, have friends use **Join IP** with your IP and port 2456. Some players report the server list is unreliable, so direct connect is often used. Also, after forwarding ports, test them using online port checkers (while server is running). **Disconnects or “incompatible version” errors:** Ensure server and clients are on the same Valheim version (after patches, everyone including the server must update). If using mods, that message often means a mod version mismatch or missing mod.
- Performance:** Valheim is fairly light on CPU but can use a lot of RAM over time (especially with many

structures built). If you notice increasing memory usage, an occasional restart can help. The server auto-saves the world periodically (default 20m). You can trigger a save via console (`save` command) if you enabled server console (`-console`). **World file issues:** A world can become corrupted (e.g., due to crash during save). If the server won't start the world (check log for world load errors), restore from the latest `.db.old` backup (rename it to `.db`). Always keep backups, especially before major updates that could alter world data. Finally, keep an eye on the **LogOutput.log** for any warnings – it might show, for example, if the server couldn't bind to the port (if so, maybe another instance is already running or port is in use).

Other Games and Notes

- **Conan Exiles:** Uses port 7777 UDP (by default) for game, similar to ARK and other UE4 games. Query port on 27015 UDP. Config via `ServerSettings.ini`. Mods are supported via the Steam Workshop – players must have the same modlist. If enabled, the server can advertise required mods and even use automatic mod downloading (the `ModList.txt` system).
- **Don't Starve Together:** Uses UDP 10999 by default. Often requires setting up a cluster (Master/ Caves servers). Config via `cluster.ini` and `server.ini` in the cluster folders. Ensure 10999 (and 10998, etc. for additional shards) are open.
- **Factorio:** Default port 34197 UDP. Config in `server-settings.json`. Very lightweight server; supports mods (all players need same mods). Check `factorio-current.log` for issues.
- **Project Zomboid:** Default UDP 16261 (plus 16262+ for additional players). Config in `Server/<servername>.ini` and `SandboxVars.lua`. Make sure to forward a range if lots of players (PZ uses one port per player slot starting at 16261).
- **Unturned:** Defaults to UDP 27015 and 27016 (one for gameplay, one for VAC). Some documentation also references 25444. Config via `Commands.dat`. Open both ports. Use `-WorkshopDownloadConfig.json` for mods from Workshop.
- **TeamSpeak / Voice Servers:** (Not a game, but sometimes alongside) Default 9987 UDP (for TS3 voice). Ensure to open UDP if hosting a voice server for your players.

(The above reference covers a broad range of popular dedicated-server games. For each game, ensure you consult the latest official documentation or community wiki for any updates in configurations or port usage. The details above preserve general defaults and known practices as of latest updates ³¹ ²¹ , but games do receive patches that might alter features.)

¹ ⁶ **Server: serverconfig.xml - 7 Days to Die Wiki**
https://7daystodie.fandom.com/wiki/Server:_serverconfig.xml

² **How To Connect To 7 Days To Die Server | Sparked Host ...**
<https://help.sparkedhost.com/en/article/how-to-connect-to-7-days-to-die-server-xguk8c/>

³ ⁴ ⁵ **Where to find log of server :: 7 Days to Die Support & Bug Reports**
<https://steamcommunity.com/app/251570/discussions/1/1734336452587854805/>

⁷ ⁸ ⁹ ¹⁰ ¹¹ **Server configuration - ARK: Survival Evolved Wiki**
https://ark.fandom.com/wiki/Server_configuration

¹² ¹⁴ ¹⁵ **Dedicated server setup - ARK: Survival Evolved Wiki**
https://ark.fandom.com/wiki/Dedicated_server_setup

13 ARK: Survival Evolved Admin Log - How to Monitor Your Server | NITRADO

<https://server.nitrado.net/en-US/guides/ark-admin-log>

16 Port Forwarding : r/arma

https://www.reddit.com/r/arma/comments/1dslylh/port_forwarding/

17 18 Can't Get Arma 3 Linux to generate RPT file - Brainless Technologies Ltd.

<https://www.brainless.us/forum/viewtopic.php?t=1829>

19 HLDS Counter Strike 1.6 Server : 13 Steps - Instructables

<https://www.instructables.com/HLDS-Counter-Strike-1.6-Server/>

20 21 22 HLDS problem please help! :: Half-Life General Discussions

<https://steamcommunity.com/app/70/discussions/0/1697169163408765282/>

23 24 25 26 Source Dedicated Server - Valve Developer Community

https://developer.valvesoftware.com/wiki/Source_Dedicated_Server

27 server.properties – Minecraft Wiki

<https://minecraft.fandom.com/wiki/Server.properties>

28 SpeedGuide.net :: Ports to Scan

https://speedguide.net/ports_sg.php?page=46&sort=port&seek=

29 30 33 Creating a server - Rust Wiki

<https://wiki.facepunch.com/rust/Creating-a-server>

31 Rust Server Ports Explained (2025) – Default, RCON, Query & Setup Guide

<https://www.eugamehost.com/blog/rust-server-ports-complete-guide-to-default-ports-setup-fixes-2025/>

32 How To Setup Logging For Your Rust Server

<https://cybrancee.com/learn/knowledge-base/how-to-setup-logging-for-your-rust-server/>

34 Killing Floor 2 | LinuxGSM

<https://docs.linuxgsm.com/game-servers/killing-floor-2>

35 UT99 Dedicated Server Connecting? : r/unrealtournament - Reddit

https://www.reddit.com/r/unrealtournament/comments/2j11sk/ut99_dedicated_server_connecting/

36 How to create an internet server? :: Unreal Tournament: Game of the ...

<https://steamcommunity.com/app/13240/discussions/0/846946775287232755?l=russian>

37 A Guide to Dedicated Servers - Valheim

<https://www.valheimgame.com/support/a-guide-to-dedicated-servers/>