Schedule txAdmin restarts with warnings and a save checkpoint

A scheduled restart should give players warning and leave a known recovery point. txAdmin supports scheduled restarts, but the existence of a countdown does not prove that a framework saved character, inventory or economy data. Configure the schedule and persistence checks as separate responsibilities.

Choose one scheduler

Check the game panel, txAdmin and any framework resources for existing restart jobs. Record their times and timezones. Select one system to own the restart; disable only a duplicate job you understand and have permission to change. Two schedulers can interrupt each other or restart during an unfinished backup.

Confirm the installed txAdmin version and your administrator permissions. The upstream project lists scheduled restarts and warnings as features. Exact setting labels vary by release. Use the schedule controls shown in your installation rather than pasting an outdated configuration block.

Build the maintenance sequence

1. Select a quiet time and publish it with an explicit timezone. Check daylight-saving changes when your staff and hosting region use different clocks.
2. Enable player warnings. Tell players when to stop trades, purchases and other actions that could remain incomplete at shutdown.
3. Read your framework's documented save procedure. Confirm which resources persist on shutdown and which require an explicit supported checkpoint. There is no universal command that safely saves every FiveM economy.
4. Schedule a consistent backup with enough time to finish before the restart. Where a stopped backup is required, make that maintenance process explicit rather than pretending it can run safely inside a short countdown.

5. Configure one initial restart and have a staff member observe it. Do not schedule frequent automatic restarts to mask an unresolved resource crash.

Use events only with reviewed code

Developers can inspect txAdmin's documented events. The scheduled-restart event supplies remaining time, while the shutdown event supplies a delay before termination. These are notifications, not a guarantee that an asynchronous database write completed.

If your framework integrates with those events, test the exact installed integration on staging. Do not paste a generic event handler that stops every resource in arbitrary order. A handler that starts a long export immediately before termination may produce a partial backup while appearing to run normally.

Verify the first scheduled restart

Check that warnings appear to a normal player, the server stops once and it starts at the expected time. Compare the console and txAdmin action log with the schedule. Reconnect and inspect a known test character's position, inventory and other agreed persistent state. Verify the backup completed independently of the successful restart.

If the timing or saved state is wrong, suspend the new schedule and use a supervised maintenance window while investigating. Restore the previous scheduler setting if necessary, retaining logs and backups. Do not repeatedly roll the database backward while players reconnect. Agree any recovery point first and keep the failed run's timestamps for framework support.

Related help

- Give FiveM staff limited txAdmin access
- Schedule game server backups and check retention
- Trace FiveM script errors and measure resource performance

Revision #1
Created 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:42 UTC by CloudExa Hosting Admin