Check ESX Legacy dependencies and resource startup

ESX Legacy is a collection of resources, not just a folder called `es_extended`. Its core version, supporting libraries, database schema and additional scripts must agree. This guide helps check an existing installation or prepare a separate test setup. It is not permission to run a fresh-install recipe over a live roleplay server.

Identify the installation you have

Record the ESX release or commit, FXServer artifact and database adapter release. Read the official core repository and the configuration shipped with that release. Save a backup of all resources and configuration plus a matching database export before changing core files. Keep a list of custom modifications, especially inventory, identity, multicharacter and jobs integrations.

If you are comparing a txAdmin recipe, inspect its actions before running it. Recipes can create or import databases and replace files. Use a new isolated directory and database for a trial deployment. Existing characters, money and inventories are not safe merely because a recipe says its installation succeeded.

Check dependencies before add-ons

The official configuration checked in November 2026 starts `oxmysql`, then `esx_lib`, then `es_extended`, followed by the remaining core resources. Treat that as release-specific reference, not a guarantee that every older ESX package has identical dependencies. Inspect your release's manifest and documentation.

```
# Simplified dependency order, not a complete configuration.
ensure oxmysql
ensure esx_lib
ensure es_extended
ensure [core]
```

Keep required settings, including the private database connection and the release's OneSync configuration, before resource startup. Preserve exact resource names and case. Do not rename a library to make its folder look tidier: imports and exports may reference that name. If a category also contains a resource already ensured above, review the effective startup carefully rather than adding more duplicate lines.

Read the first boot error

- **Database connection failure:** establish the adapter's connection before investigating character scripts.
- **Missing table or column:** compare installed schema and migration history with the matching ESX release. Do not recreate the production database.
- **Missing export or library:** confirm the named dependency started successfully and is the expected version.
- **Old shared-object event errors:** identify the add-on's supported ESX integration. Use the author's current migration instructions; do not paste a guessed replacement into every resource.

Disable a newly added optional script only during planned maintenance, leaving the known working core unchanged. Restarting ESX beneath running jobs and inventory scripts can leave inconsistent state. Prefer a controlled full restart after saving player data.

Test the complete player path

Join with a test account, reach character selection or creation, spawn and perform one ordinary inventory or job action. Disconnect normally, reconnect and confirm the expected character state persists. Test existing characters as well as new ones: a schema mismatch may affect only old records. Keep tests small and avoid commands that give money or items to real players.

For support, provide the first redacted startup error, exact ESX/core versions, adapter version and recent resource changes. Remove licence keys, connection strings and player identifiers. A functioning database connection alone does not prove ESX character loading or an add-on's compatibility.

Related guides

- Connect oxmysql to your database
- Check manifests and resource dependencies

Further reading

- ESX Legacy official core repository
- ESX reference configuration
- ESX core resource manifest

Looking for another game? Browse game server setup and troubleshooting guides.

Revision #2
Created 2026-11-09 09:21:20 UTC by CloudExa Hosting Admin
Updated 2026-11-09 09:21:21 UTC by CloudExa Hosting Admin