

atloria

September 4, 2026

atloria

Kind: Module

Source: `atloria.sln`

.NET solution file

`atloria.sln` is the .NET solution file that groups the Atloria projects into a single build, test, and development workspace. It provides the entry point for IDEs and the .NET CLI to restore dependencies, compile projects, and run solution-level commands.

Diagram

```
graph TD
    S[atloria.sln<br/>.NET Solution] --> P1[Application Projects]
    S --> P2[Library Projects]
    S --> P3[Test Projects]

    P1 --> B1[dotnet build]
    P2 --> B2[dotnet build]
    P3 --> T[dotnet test]
```

Usage

```
import { execSync } from "node:child_process";

// Restore NuGet dependencies for every project in the solution.
execSync("dotnet restore atloria.sln", { stdio: "inherit" });

// Build all projects using the Release configuration.
execSync("dotnet build atloria.sln --configuration Release --no-restore", {
  stdio: "inherit",
});

// Run all tests registered in the solution.
execSync("dotnet test atloria.sln --configuration Release --no-build", {
```

```
    stdio: "inherit",  
  });
```

AI Coding Instructions

- Add new .NET projects to `atloria.sln` so they are included in solution-wide builds and test runs.
- Use `dotnet build atloria.sln` and `dotnet test atloria.sln` to validate cross-project dependency changes.
- Run `dotnet restore` before building in clean environments; use `--no-restore` only when dependencies are already restored.
- Keep project references defined in individual `.csproj` files rather than relying on solution-folder organization.

Relationships

- MODULE_DECLARES → `atloria-monorepo`
- MODULE_DECLARES → `libs`
- MODULE_DECLARES → `parser-core`
- MODULE_DECLARES → `dotnet-bridge`
- MODULE_DECLARES → `Atloria.CSharpParser`