

System.Xml.Linq

August 18, 2026

System.Xml.Linq

Kind: Service

Source: Pams/Core/Pams.Core/Pams.Core.csproj (line 1)

Part of: Pams

NuGet package dependency

System.Xml.Linq is a NuGet package dependency declared by Pams/Core/Pams.Core/Pams.Core.csproj . It lets Pams.Core C# code read, query, create, and write XML through LINQ to XML types such as XDocument and XElement . NuGet restores the dependency during the Pams.Core build so XML-related code can reference these types.

Diagram

```
sequenceDiagram
    participant Developer
    participant MSBuild
    participant NuGet
    participant PamsCore as Pams.Core
    participant XmlApi as System.Xml.Linq

    Developer->>MSBuild: Build Pams.Core project
    MSBuild->>NuGet: Restore package dependency
    NuGet-->>MSBuild: Return package assets
    MSBuild->>PamsCore: Compile C# source
    PamsCore->>XmlApi: Read or create XML with XDocument and XElement
```

Usage

```
import { execFileSync } from "node:child_process";
import { resolve } from "node:path";
```

```

const projectFile = resolve(
  "Pams/Core/Pams.Core/Pams.Core.csproj",
);

// Restore System.Xml.Linq and other project dependencies.
execFileSync("dotnet", ["restore", projectFile], {
  stdio: "inherit",
});

// Build the C# project that references System.Xml.Linq.
execFileSync("dotnet", ["build", projectFile], {
  stdio: "inherit",
});

```

AI Coding Instructions

- Keep the `System.Xml.Linq` package reference in `Pams/Core/Pams.Core/Pams.Core.csproj` when code depends on `XDocument`, `XElement`, or related XML types.
- Write XML handling in C#; TypeScript code should invoke the built .NET project rather than import this NuGet package.
- Run `dotnet restore` before building when package assets are not present locally.
- Prefer LINQ to XML types for XML parsing and updates instead of string-based XML edits.