

System

August 18, 2026

System

Kind: Service

Source: Pams/Logic/Pams.Email/Pams.Email.csproj (line 1)

Part of: Pams

NuGet package dependency

Pams.Email is a legacy-format project that declares NuGet package dependencies for the email component. It participates in the PAMS build by restoring referenced packages before the email project is compiled and consumed by other applications.

Diagram

```
sequenceDiagram
    participant Developer
    participant Build as Build Process
    participant Email as Pams.Email.csproj
    participant NuGet
    participant App as PAMS Application

    Developer->>Build: Run restore and build
    Build->>Email: Read project dependencies
    Email->>NuGet: Request referenced packages
    NuGet-->>Build: Restore packages
    Build->>Email: Compile email project
    Email-->>App: Provide compiled email component
```

Usage

```
import { execFile } from "node:child_process";
import { promisify } from "node:util";

const execFileAsync = promisify(execFile);
```

```

async function restoreEmailProject() {
  const projectPath = "Pams/Logic/Pams.Email/Pams.Email.csproj";

  await execFileAsync("dotnet", ["restore", projectPath]);
  await execFileAsync("dotnet", ["build", projectPath, "--no-restore"]);

  console.log("Pams.Email dependencies restored and project built.");
}

restoreEmailProject().catch((error) => {
  console.error("Failed to build Pams.Email:", error);
  process.exitCode = 1;
});

```

AI Coding Instructions

- Keep NuGet package references in `Pams.Email.csproj` aligned with the legacy project format already used by the project.
- Run `dotnet restore` before building when package references or package sources change.
- Avoid changing package versions without checking projects that reference the compiled `Pams.Email` output.
- Update package references in the project file rather than adding JavaScript or TypeScript package management files to this .NET project.