

Microsoft.Extensions.Logging.Debug

August 17, 2026

Microsoft.Extensions.Logging.Debug

Kind: Service

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

Part of: Pams

NuGet package dependency

Microsoft.Extensions.Logging.Debug is a NuGet dependency declared by Pams.Web.csproj . It adds the .NET debug logging provider, allowing application log entries to be written to the debugger output when the web project configures debug logging.

Diagram

```
sequenceDiagram
    participant App as Pams.Web
    participant Logging as ILogger
    participant Debug as Debug Logger Provider
    participant Output as Debug Output

    App->>Logging: LogInformation(message)
    Logging->>Debug: Write log entry
    Debug->>Output: Emit formatted message
```

Usage

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

const app = spawn("dotnet", ["run", "--project", "Pams/Web/Pams.Web/Pams.Web.csproj"], {
  env: {
    ...process.env,
    Logging__LogLevel__Default: "Debug",
  },
  stdio: "inherit",
});
```

```
app.on("exit", (code) => {  
    console.log(`Pams.Web exited with code ${code}`);  
});
```

AI Coding Instructions

- Keep `Microsoft.Extensions.Logging.Debug` declared as a package dependency in `Pams.Web.csproj` when debug logger output is required.
- Configure the provider in the .NET application startup code with `AddDebug()` before expecting debug output.
- Use `ILogger<T>` through dependency injection rather than creating logger instances directly.
- Check the active log-level configuration when debug messages do not appear; `Debug` entries can be filtered by application settings.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Injected or called by (1)

- `Pams.Web` — `Pams/Web/Pams.Web/Pams.Web.csproj` :1