. 2025년 10월 30일 MAUI프로젝트 Framework .NET 8.0을 생성하니까 NETSDK1202 오류 발행

This error is happening because the **.NET MAUI support policy is shorter** than the general .NET policy. While .NET 8 is a Long-Term Support (LTS) release, the MAUI workloads for a specific .NET version are only supported for **6 months** after the next major MAUI version ships.

Based on the MAUI support policy:

* **MAUI 8** (which corresponds to net8.0-\* workloads) was released in November 2023.
* The next major version, **MAUI 9** (corresponding to net9.0-\*), shipped in November 2024.
* MAUI 8 then receives support for **6 months** after MAUI 9's release.
* Therefore, **MAUI 8's support ended around May 2025**.

Since the current date is October 2025, your .NET 8 MAUI workload for iOS (net8.0-ios) is now past its end-of-life, which triggers the **NETSDK1202** error when you use a newer .NET SDK that enforces this check (likely a recent **.NET 9 SDK**).

**🛠️ Solutions to Fix NETSDK1202**

You have two main paths to resolve this error: the **recommended solution** (upgrade) or the **temporary workaround** (disable the check).

**1. ✅ Recommended Solution: Upgrade to .NET 9 MAUI**

The most secure and future-proof way to fix this is to upgrade your project to the current supported MAUI version, which is **.NET MAUI 9**.

* **Update your Target Framework:** In your project file (.csproj), change the Target Framework Monikers (TFMs) from net8.0-\* to net9.0-\*.

XML

<PropertyGroup>

<TargetFrameworks>net9.0-android;net9.0-ios;net9.0-maccatalyst;net9.0-windows10.0.19041.0</TargetFrameworks>

</PropertyGroup>

* **Update NuGet Packages:** Make sure all of your Microsoft.Maui.\* NuGet packages are also updated to the latest 9.x version.
* **Install .NET 9 SDK:** Ensure you have the latest .NET 9 SDK installed.

**2. ⏳ Temporary Workaround: Use a global.json File**

If you are blocked from upgrading to .NET 9, you may be using a newer SDK (e.g., .NET 9 SDK) to build an older target (e.g., net8.0-ios), and the newer SDK is what's enforcing the check. You can pin your project to use a specific, older .NET 8 SDK version to avoid the check.

* **Create a global.json file** in the root of your solution folder.
* **Pin a compatible .NET 8 SDK version** (one that was released *before* the workload was marked EOL, or one that is simply less strict).

JSON

{

"sdk": {

"version": "8.0.300",

"rollForward": "latestFeature"

}

}

(Note: You may need to experiment to find a specific .NET 8 SDK version that still allows the build.)

**3. ⚠️ Non-Recommended Workaround: Disable the Check**

If you need to quickly get a build working and understand the security implications, you can disable the end-of-life check entirely.

* **Edit your project file** (.csproj).
* **Add the following property** to a <PropertyGroup>:

XML

<PropertyGroup>

<CheckEolWorkloads>false</CheckEolWorkloads>

</PropertyGroup>

**Warning:** This allows the project to build but leaves you on an unsupported platform that will not receive security or stability updates, and it may eventually fail to build due to conflicts with the latest Xcode or Android SDKs.

**Would you like me to search for the specific steps to safely upgrade a .NET MAUI project from .NET 8 to .NET 9?**