

Mastering MSBuild:

Troubleshooting, Customization, and more

by Jan Krivanek & Yuliia Kovalova



Agenda

Introduction to MSBuild

MSBuild.exe vs MSBuild.dll

Troubleshooting

Advanced Customization

Introduction to MSBuild





What is MSBuild?

- Declarative, data driven, XML based programming *language*
- A battery of predefined *scripts* defining the build behaviour
- Can be hard to troubleshoot

REPOSITORY: [dotnet/msbuild: The Microsoft Build Engine \(MSBuild\) is the build platform for .NET and Visual Studio.](https://github.com/dotnet/msbuild)

Key Characteristics

- **Extensible/flexible build process**
- **Part of .NET SDK, Visual Studio ecosystem, and C# DevKit**
- **Fundamental to .NET development**



MSBuild Components - VS and SDK Integration

Integration with Visual Studio

- MSBuild.exe is a build tool for VS
- Integrated with Visual Studio Installer
- Follows VS release lifecycle



```
PS C:\Users\ykovalova> C:\Program Files\Microsoft Visual Studio\2022\{Edition}\MSBuild\Current\Bin\MSBuild.exe
```

dotNET SDK

- Contains MSBuild.dll
- Executes it in dotnet CLI (dotnet build)
- Why? - Cross platform capabilities!
- .NET Release lifecycle



```
# .NET SDK Path
C:\Program Files\dotnet\sdk\{version}\

# MSBuild.dll Location
C:\Program Files\dotnet\sdk\{version}\MSBuild.dll
```


SDK vs Legacy Project Files

```
repos > checkLocator > checkLocator > checkLocator.csproj
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <OutputType>Exe</OutputType>
5     <TargetFramework>net9.0</TargetFramework>
6     <ImplicitUsings>enable</ImplicitUsings>
7     <Nullable>enable</Nullable>
8   </PropertyGroup>
9
10 </Project>
11
```

SDK

```
DiscoveryHelper.csproj X
repos > ConsoleApp8 > ConsoleApp8 > DiscoveryHelper.csproj
1 <?xml version="1.0" encoding="utf-8"?>
2 <Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3   <Import Project="..\packages\EntityFramework.6.4.4\build\EntityFramework.props" Condition="Exists('..\packages\EntityFramework.6.4.4\build\Enti
4   <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildT
5   <PropertyGroup>
6     <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
7     <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
8     <ProjectGuid>{F377407F-E53E-4DB3-8BC6-67A6F803D70E}</ProjectGuid>
9     <OutputType>Exe</OutputType>
10    <RootNamespace>ConsoleApp8</RootNamespace>
11    <AssemblyName>ConsoleApp8</AssemblyName>
12    <TargetFrameworkVersion>v4.7.2</TargetFrameworkVersion>
13    <FileAlignment>512</FileAlignment>
14    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
15    <Deterministic>true</Deterministic>
16    <NuGetPackageImportStamp>
17    </NuGetPackageImportStamp>
18  </PropertyGroup>
19  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
20    <PlatformTarget>AnyCPU</PlatformTarget>
21    <DebugSymbols>true</DebugSymbols>
22    <DebugType>full</DebugType>
23    <Optimize>>false</Optimize>
24    <OutputPath>bin\Debug\</OutputPath>
25    <DefineConstants>DEBUG;TRACE</DefineConstants>
26    <ErrorReport>prompt</ErrorReport>
27    <WarningLevel>4</WarningLevel>
28  </PropertyGroup>
29  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
30    <PlatformTarget>AnyCPU</PlatformTarget>
31    <DebugType>pdbonly</DebugType>
32    <Optimize>true</Optimize>
33    <OutputPath>bin\Release\</OutputPath>
34    <DefineConstants>TRACE</DefineConstants>
35    <ErrorReport>prompt</ErrorReport>
36    <WarningLevel>4</WarningLevel>
37  </PropertyGroup>
```

Legacy Project file (none-sdk)

Bright Future

Unification of SDK CLI and VS build.

Steps:

- 1. Context – guaranteeing same env vars**
- 2. Core Task host – ability to run core tasks from Visual Studio**
- 3. (never :-)) VS core (use .Net SDK as a basement)**

MSBuild Depth



MSBuild Core Concepts

- Project Files (e.g. .csproj, .fsproj)
- Properties
- Items
- Targets
- Tasks

```
Demo.csproj
repos > checkLocator > checkLocator > Demo.csproj
1  <Project Sdk="Microsoft.NET.Sdk">
2
3      <!-- Property Group: Configuration and Project-wide Settings -->
4      <PropertyGroup>
5          <TargetFramework>net8.0</TargetFramework>
6          <OutputType>Exe</OutputType>
7          <Version>1.0.0</Version>
8          <Configuration Condition="'$(Configuration)' == ''>Debug</Configuration>
9      </PropertyGroup>
10
11     <!-- Item Groups: Project Resources and References -->
12     <ItemGroup>
13         <!-- Package References -->
14         <PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
15         <PackageReference Include="Serilog" Version="3.1.1" />
16     </ItemGroup>
17
18     <!-- Custom Targets -->
19     <Target Name="PreBuild" BeforeTargets="PrepareForBuild">
20         <Message Text="Starting pre-build process" Importance="high" />
21         <MakeDir Directories="$(BuildOutputPath)" />
22     </Target>
23
24     <Target Name="CustomDeployment" AfterTargets="Build">
25         <Message Text="Custom deployment process" Importance="high" />
26
27         <!-- Built-in Task -->
28         <Copy
29             SourceFiles="@(\DeploymentFiles)"
30             DestinationFolder="$(OutputPath)\deployment"
31         />
32     </Target>
33 </Project>
```

MSBuild Concepts vs Programming Language

- Properties = fields

Demo.csproj •

repos > checkLocator > checkLocator > Demo.csproj

```
1  <Project Sdk="Microsoft.NET.Sdk">
2
3      <!-- Property Group: Configuration and Project-wide Settings -->
4      <PropertyGroup>
5          <TargetFramework>net8.0</TargetFramework>
6          <OutputType>Exe</OutputType>
7          <Version>1.0.0</Version>
8          <Configuration Condition="'$(Configuration)' == ''>Debug</Configuration>
9      </PropertyGroup>
10
```

MSBuild Concepts vs Programming Language

- Items = Dictionary

```
10
11 <!-- Item Groups: Project Resources and References -->
12 <ItemGroup>
13   <!-- Package References -->
14   <PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
15   <PackageReference Include="Serilog" Version="3.1.1" />
16 </ItemGroup>
17
```

MSBuild Concepts vs Programming Language

- Targets = class/struct

```
17
18      <!-- Custom Targets -->
19      <Target Name="PreBuild" BeforeTargets="PrepareForBuild">
20          <Message Text="Starting pre-build process" Importance="high" />
21          <MakeDir Directories="$(BuildOutputPath)" />
22      </Target>
23
```

MSBuild Concepts vs Programming Language

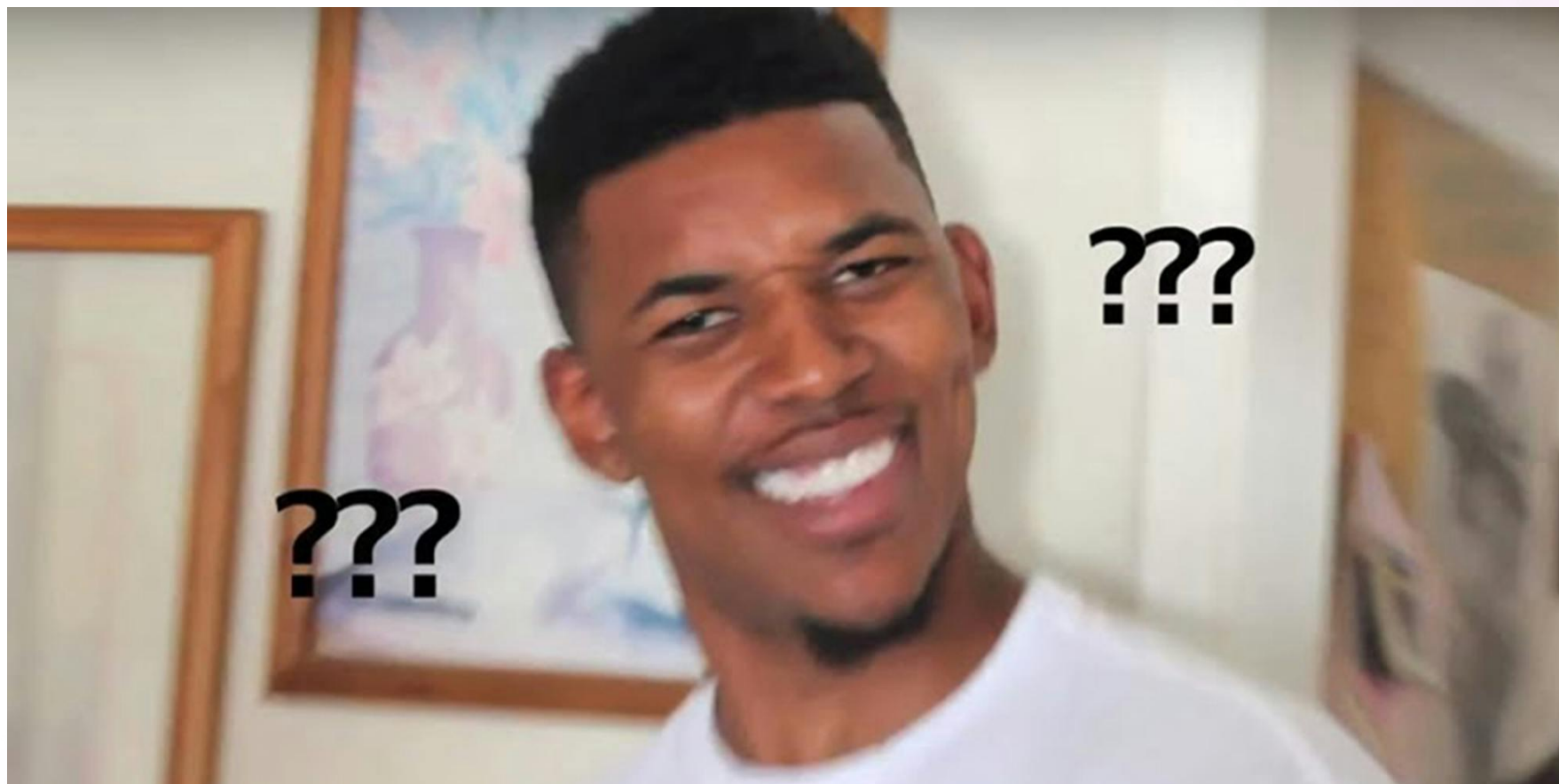
- Tasks = functions/methods

```
24 <Target Name="CustomDeployment" AfterTargets="Build">
25   <Message Text="Custom deployment process" Importance="high" />
26
27   <!-- Built-in Task -->
28   <Copy
29     SourceFiles="@DeploymentFiles"
30     DestinationFolder="$(OutputPath)\deployment"
31   />
32 </Target>
```


Build Workflow

1. Project Loading (Parsing xml)
2. **Property** Evaluation (Reading Property values)
3. **Item** Transformation (Pulling references)
4. **Target** Evaluation
5. **Task** Execution
6. Post-Build Processing
7. Logging and Reporting

```
Demo.csproj
repos > checkLocator > checkLocator > Demo.csproj
1  <Project Sdk="Microsoft.NET.Sdk">
2
3  <!-- Property Group: Configuration and Project-wide Settings -->
4  <PropertyGroup>
5    <TargetFramework>net8.0</TargetFramework>
6    <OutputType>Exe</OutputType>
7    <Version>1.0.0</Version>
8    <Configuration Condition="'$(Configuration)' == ''>Debug</Configuration>
9  </PropertyGroup>
10
11 <!-- Item Groups: Project Resources and References -->
12 <ItemGroup>
13   <!-- Package References -->
14   <PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
15   <PackageReference Include="Serilog" Version="3.1.1" />
16 </ItemGroup>
17
18 <!-- Custom Targets -->
19 <Target Name="PreBuild" BeforeTargets="PrepareForBuild">
20   <Message Text="Starting pre-build process" Importance="high" />
21   <MakeDir Directories="$(BuildOutputPath)" />
22 </Target>
23
24 <Target Name="CustomDeployment" AfterTargets="Build">
25   <Message Text="Custom deployment process" Importance="high" />
26
27 <!-- Built-in Task -->
28 <Copy
29   SourceFiles="@(\DeploymentFiles)"
30   DestinationFolder="$(OutputPath)\deployment"
31 />
32 </Target>
33 </Project>
```



MSBuild Binlog

/bl[:name.binlog]

\$env:MSBUILDDDEBUGENGINE = 1



msbuildlog.com (thank you **Kirill Osenkov!**)



Demo: Structured BinLog Viewer



Collecting binlogs in VS

cmd:

```
> SET MSBUILDDEBUGENGINE=1  
> SET MSBUILDDEBUGPATH=C:\MSBuildReproLogs  
> devenv.exe MySolution.sln
```

MSBuild relies on env variables

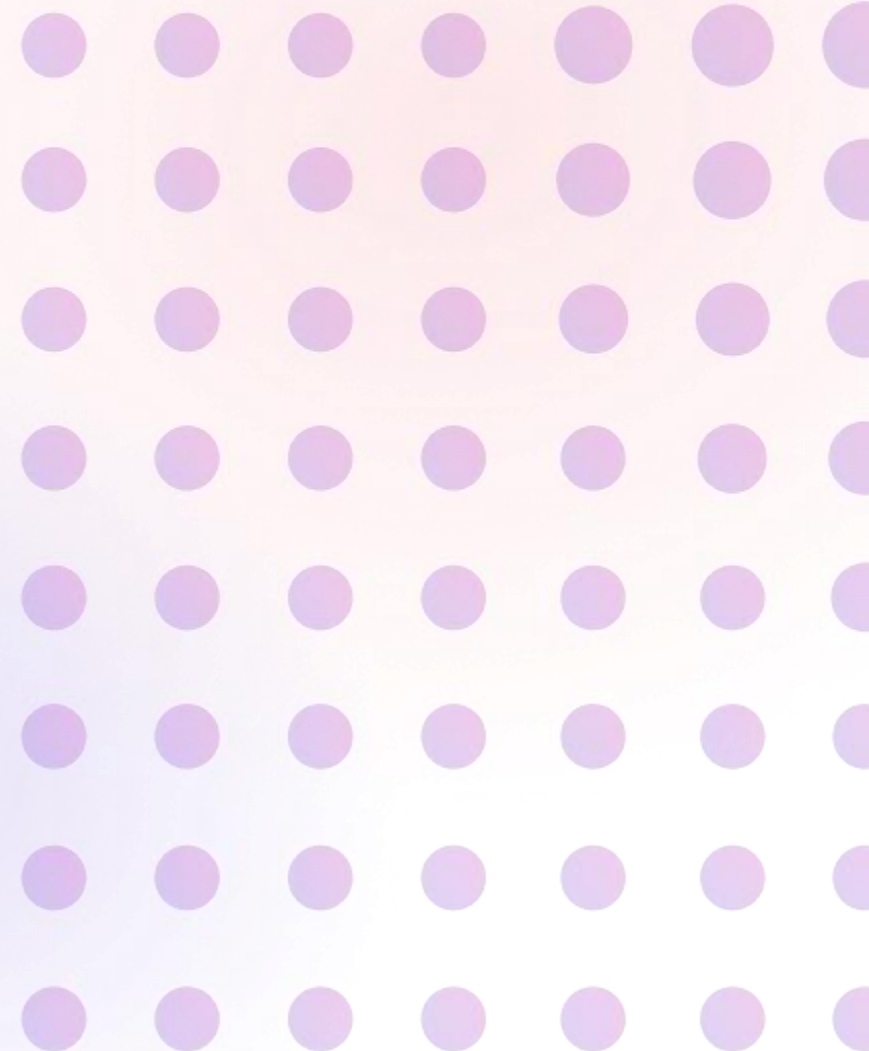
Build Waves and Feature Flags/Traits

- Control build behavior
- Enable/disable experimental features
- Get extra information in logs

[SdkAnalysisLevel](#) (is getting more attention!)

[ChangeWaves.md](#)

[MSBuild-Environment-Variables.md](#)



MSBuild Customization



MSBuild Extensibility

NuGet Packages

Directory.Build.props/targets (e.g. [hidden.bomb.targets](#))

.user, .rsp files

MSBuildExtensionsPath & MSBuildUserExtensionsPath



Demo: MSBuild Extensibility

Conclusions

- G LÀ° sũ ÄÄvÈ ì yk ưũ Äi Ägũ
- J yv y Ä sũv zä g z Ä y Ä y° g sũ Ä sũ Fv c Ä s È ì y Ä Ä v° y Ä y s ư 7
- Ä r ì gũ Ä v g° ư ì ư ä v ư Ä y Ä y v Ä z° ì z

Ò ì Ä È è Ä È ì g v ư ì Ä v ư ä y g° ä v ư z Ä v Ä° y Ä y v 8 Ä
ř g y z 8 9 9 s ấ ° g g v ư 9 v ä ư ä 9 z g° sũ



Thank you