

概述

编译选项可以在使用MSBuild的项目里通过MSBuild属性指定。

例子

```
<PropertyGroup Condition="'$(Configuration)' == 'Debug'">
  <TypeScriptRemoveComments>false</TypeScriptRemoveComments>
  <TypeScriptSourceMap>true</TypeScriptSourceMap>
</PropertyGroup>
<PropertyGroup Condition="'$(Configuration)' == 'Release'">
  <TypeScriptRemoveComments>true</TypeScriptRemoveComments>
  <TypeScriptSourceMap>false</TypeScriptSourceMap>
</PropertyGroup>
<Import
  Project="$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)\Microsoft.Common.props" Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)\Microsoft.Common.props')>
```

映射

编译选项	MSBuild属性名称	
--allowJs	MSBuild 不支持此选项	
--allowSyntheticDefaultImports	TypeScriptAllowSyntheticDefaultImports	布尔值
--allowUnreachableCode	TypeScriptAllowUnreachableCode	布尔值
--allowUnusedLabels	TypeScriptAllowUnusedLabels	布尔值
--alwaysStrict	TypeScriptAlwaysStrict	布尔值
--baseUrl	TypeScriptBaseUrl	文件路径
--charset	TypeScriptCharset	
--declaration	TypeScriptGeneratesDeclarations	布尔值
--declarationDir	TypeScriptDeclarationDir	文件路径
--diagnostics	MSBuild 不支持此选项	
--disableSizeLimit	MSBuild 不支持此选项	
--emitBOM	TypeScriptEmitBOM	布尔值
--emitDecoratorMetadata	TypeScriptEmitDecoratorMetadata	布尔值
--experimentalAsyncFunctions	TypeScriptExperimentalAsyncFunctions	布尔值
--experimentalDecorators	TypeScriptExperimentalDecorators	布尔值
--forceConsistentCasingInFileNames	TypeScriptForceConsistentCasingInFileNames	布尔值
--help	MSBuild 不支持此选项	
--importHelpers	TypeScriptImportHelpers	布尔值
--inlineSourceMap	TypeScriptInlineSourceMap	布尔值
--inlineSources	TypeScriptInlineSources	布尔值
--init	MSBuild 不支持此选项	
--isolatedModules	TypeScriptIsolatedModules	布尔值

--jsx	TypeScriptJSXEmit	React或Pres
--jsxFactory	TypeScriptJSXFactory	有效的名字
--lib	TypeScriptLib	逗号分隔的
--listEmittedFiles	MSBuild 不支持此选项	
--listFiles	MSBuild 不支持此选项	
--locale	automatic	自动设置为I
--mapRoot	TypeScriptMapRoot	文件路径
--maxNodeModuleJsDepth	MSBuild 不支持此选项	
--module	TypeScriptModuleKind	AMD， Common
--moduleResolution	TypeScriptModuleResolution	Classic或No
--newLine	TypeScriptNewLine	CRLF或LF
--noEmit	MSBuild 不支持此选项	
--noEmitHelpers	TypeScriptNoEmitHelpers	布尔值
--noEmitOnError	TypeScriptNoEmitOnError	布尔值
--noFallthroughCasesInSwitch	TypeScriptNoFallthroughCasesInSwitch	布尔值
--noImplicitAny	TypeScriptNoImplicitAny	布尔值
--noImplicitReturns	TypeScriptNoImplicitReturns	布尔值
--noImplicitThis	TypeScriptNoImplicitThis	布尔值
--noImplicitUseStrict	TypeScriptNoImplicitUseStrict	布尔值
--noUnusedLocals	TypeScriptNoUnusedLocals	布尔值
--noUnusedParameters	TypeScriptNoUnusedParameters	布尔值
--noLib	TypeScriptNoLib	布尔值
--noResolve	TypeScriptNoResolve	布尔值
--out	TypeScriptOutFile	文件路径
--outDir	TypeScriptOutDir	文件路径
--outFile	TypeScriptOutFile	文件路径
--paths	MSBuild 不支持此选项	
--preserveConstEnums	TypeScriptPreserveConstEnums	布尔值
--listEmittedFiles	MSBuild 不支持此选项	
--pretty	MSBuild 不支持此选项	
--reactNamespace	TypeScriptReactNamespace	字符串
--removeComments	TypeScriptRemoveComments	布尔值
--rootDir	TypeScriptRootDir	文件路径
--rootDirs	MSBuild 不支持此选项	
--skipLibCheck	TypeScriptSkipLibCheck	布尔值
--skipDefaultLibCheck	TypeScriptSkipDefaultLibCheck	布尔值
--sourceMap	TypeScriptSourceMap	文件路径
--sourceRoot	TypeScriptSourceRoot	文件路径
--strictNullChecks	TypeScriptStrictNullChecks	布尔值
--stripInternal	TypeScriptStripInternal	布尔值

<code>--suppressExcessPropertyErrors</code>	<code>TypeScriptSuppressExcessPropertyErrors</code>	布尔值
<code>--suppressImplicitAnyIndexErrors</code>	<code>TypeScriptSuppressImplicitAnyIndexErrors</code>	布尔值
<code>--target</code>	<code>TypeScriptTarget</code>	ES3, ES5, !
<code>--traceResolution</code>	<i>MSBuild 不支持此选项</i>	
<code>--types</code>	<i>MSBuild 不支持此选项</i>	
<code>--typeRoots</code>	<i>MSBuild 不支持此选项</i>	
<code>--watch</code>	<i>MSBuild 不支持此选项</i>	
<i>MSBuild only option</i>	<code>TypeScriptAdditionalFlags</code>	任何编译选.

我使用的Visual Studio版本里支持哪些选项？

查找 `C:\Program Files`

`(x86)\MSBuild\Microsoft\VisualStudio\v$(VisualStudioVersion)\TypeScript\Microso`
文件。

可用的MSBuild XML标签与相应的tsc编译选项的映射都在那里。

ToolsVersion

工程文件里的`<TypeScriptToolsVersion>1.7</TypeScriptToolsVersion>`属性值表明了构建时使用的编译器的版本号（这个例子里是1.7）
这样就允许一个工程在不同的机器上使用固定的版本去编译。

如果没有指定`TypeScriptToolsVersion`，则会使用机器上安装的最新版本的编译器去构建。

如果用户使用的是更新版本的TypeScript，则会在首次加载工程的时候看到一个提示升级工程的对话框。

TypeScriptCompileBlocked

如果你使用其它的构建工具（比如，gulp， grunt等等）并且使用VS做为开发和调试工具，那么在工程里设置`<TypeScriptCompileBlocked>true</TypeScriptCompileBlocked>`。
这样VS只会提供给你编辑的功能，而不会在你按F5的时候去构建。