

第101期-关于 .NET 5.0 改进与升级

2020年4月28日 14:51

.NET 5 介绍

.NET 5.0是3.1之后的.NET Core的下一个主要版本。我们将此新版本命名为.NET 5.0而不是.NET Core 4.0，主要原因：

- 我们跳过了版本号4.x，以避免与.NET Framework 4.x混淆
- 我们从名称中删除了 “Core”，以强调这是.NET未来的主要实现。

ASP.NET Core 5.0 基于 .NET 5.0，但保留名称 “Core” 以避免将其与 ASP.NET MVC 5 混淆。

Entity Framework Core 5.0 保留名称 “Core” 以避免将其与 Entity Framework 5 和 6

安装 Visual Studio 2019 16.8 后自带 .NET 5 SDK，理解 SDK 和 Runtime 的区别。

.NET 5.0 不能替代 .NET Framework 框架

Web Forms -----> ASP.NET Core Blazor or Razor Pages

Windows Workflow (WF) -----> CoreWF

Windows Communication Foundation (WCF) ----->WcfClient 或 CoreWCF 或 gRPC

.NET 5.0 不会替代 .NET Standard 标准

新的应用程序开发可以net5.0为所有项目类型（包括类库）指定目标框架别名（TFM）。在.NET 5工作负载之间共享代码得到了简化，因为您只需要 net5.0 TFM。

但是，如果您打算在.NET Framework，.NET Core和.NET 5工作负载之间共享代码，则可以通过指定netstandard2.1为TFM来实现。

C# 9 更新

有关可用的C# 9功能的更多信息，请参见 C# 9中的新增 功能。

```
var threshChoice = Console.ReadKey();
if (threshChoice.KeyChar is 'Y' or 'y')
{
    TestThresholdAndInterrupt(ccs811);
}
```

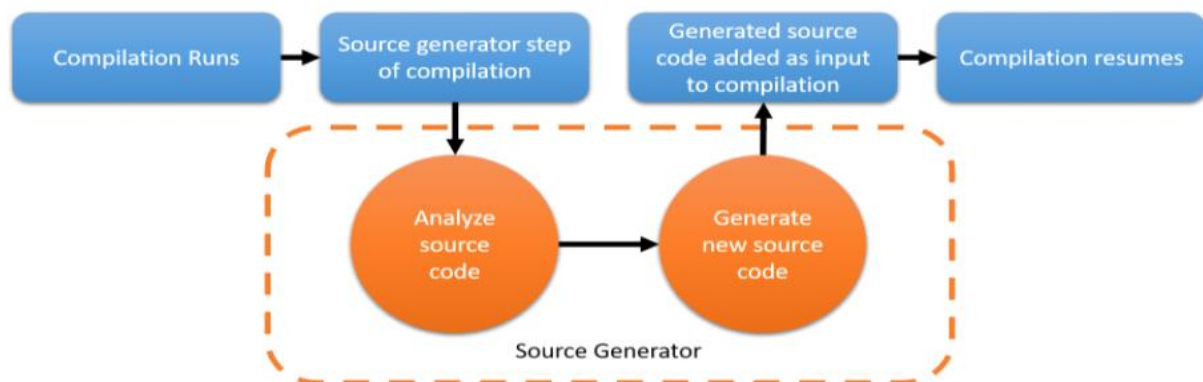
```
if (pollingType is null or { Length: >15 })  
{  
    return null;  
}
```

源代码生成器

C# 的新功能，可让C# 开发人员检查用户代码并生成可添加到编译中的新C# 源文件。这是通过一种称为源代码生成器的新型组件完成的。我们希望在.NET 6.0及更高版本中的.NET产品中更多地使用源生成器。

源代码生成器是一段在编译过程中运行的代码，可以检查您的程序以生成与其他代码一起编译的其他文件。

如果您熟悉 **Roslyn Analyzers** 分析器，则可以将 Source Generators 视为可以发出 C# 源代码的分析器，重新颠覆运行时反射，IL编译处理AOP 和 MSBuild 任务。典型的 PostSharp。在编译阶段动态注入。



[Introducing C# Source Generators](#)

[New C# Source Generator Samples](#)

[.NET Compiler Platform \(Roslyn Analyzers\)](#)

性能进一步提升

GC垃圾回收器，JIT 实时编译器，HttpClient 底层更快，JsonSerializer 更快功能更多，Roslyn 编译器更快还支持内插分析器，字符串处理和字符处理性能提升，Regex 正则表达式性能改进，线程和异步性能提升，集合和字典性能提升，LINQ 更快，以性能为中心的API设计。

[Performance Improvements in .NET 5](#)

Benchmark.NET 现在是衡量.NET代码性能的规范工具。

ARM64 Performance in .NET 5

System.Text.Json 库的新功能

保留引用并处理循环引用

```
JsonSerializerOptions options = new()
{
    ReferenceHandler = ReferenceHandler.Preserve,
    WriteIndented = true
};

string tylerJson = JsonSerializer.Serialize(obj, options);
```

HttpClient 和 HttpContent 扩展方法

```
User user = await httpClient.GetFromJsonAsync<User>("users/1");
HttpResponseMessage response = await httpClient.PostAsJsonAsync("/api/users", user);
```

允许或在引号中写数字

```
{ "DegreesCelsius": "23" } 和 { "DegreesCelsius": 23 }
```

```
JsonSerializerOptions options = new JsonSerializerOptions
{
    NumberHandling = JsonNumberHandling.AllowReadingFromString |
    JsonNumberHandling.WriteAsString,
    WriteIndented = true
};
```

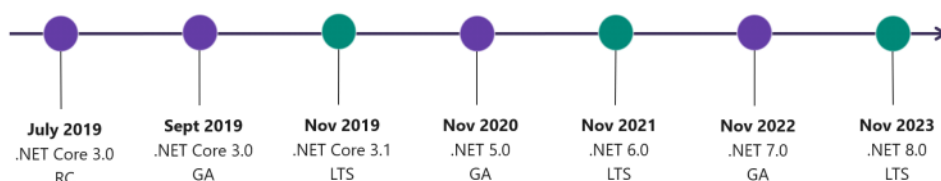
- [保留引用并处理循环引用](#)
- [HttpClient和HttpContent扩展方法](#)
- [允许或在引号中写数字](#)
- [支持不可变类型和C# 9记录](#)
- [支持非公共有参构造函数](#)
- [支持字段](#)
- [有条件地忽略属性](#)
- [支持非字符串键字典](#)
- [允许自定义转换器处理null](#)
- [复制JsonSerializerOptions](#)
- [使用Web默认创建JsonSerializerOptions](#)

发布时间和路线图

然后我们打算每年11月发布一次主要版本的.NET版本，上个版本延期三个月停止支持。我们希望在2022年2月前支持.NET5.0。 .NET6.0将是LTS版本，并且将像.NET Core 3.1一样受三年支持。

每个新版本的Windows都将继续支持.NET Framework。去年，我们宣布已停止向.NET Framework 添加新功能。 .NET Core 3.X 结束了 .NET Framework API 移植项目。

.NET Schedule



- .NET Core 3.0 release in September
- .NET Core 3.1 = Long Term Support (LTS)
- .NET 5.0 release in November 2020
- Major releases every year, LTS for even numbered releases
- Predictable schedule, minor releases if needed

Alpha、Beta、RC、GA、LTS 等软件各个版本号的含义

生态系统



参考资料

.NET 5 新增功能

.NET 5.0 中断性变更

[Introducing .NET 5](#)

[Announcing .NET 5.0](#)