

Online on-demand Project Support



WPF und .NET (Core) – Back again



Thorsten Kansy (tkansy@dotnetconsulting.eu)

Meine Person- Thorsten Kansy

Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor



Mein Service- Ihr Benefit

- Individuelle Inhouse Trainings
- (Online on-demand) Projektbegleitung
- Beratung
 - Problemanalyse und Lösungen
 - Technologieentscheidungen



Agenda

- WPF unter .NET Core
- Was fehlt?
- Windows Compatibility Pack
- Migration
- Publishing/ Deployment

.NET Core Roadmap



Update: 2023

Releases

Version	Startdatum	Enddatum
.NET 8	14. Nov. 2023	
.NET 7	8. Nov. 2022	14. Mai 2024
.NET 6.0 (LTS)	8. Nov. 2021	12. Nov. 2024
.NET 5.0	10. Nov. 2020	10. Mai 2022
.NET Core 3.1 (LTS)	3. Feb. 2019	13. Dez. 2020



<https://learn.microsoft.com/de-de/lifecycle/products/microsoft-net-and-net-core>
<https://www.youtube.com/watch?v=mna5fg7QGz8>

Zukunft von .NET Framework

- .NET Framework 4.8 ist die letzte .NET Framework Feature-Version?
 - .NET Framework 4.8.1 kam
- Migration auf 4.8 nur bei neuen Features notwendig?
 - Ja, kann man so sagen. Aber die weitere Entwicklung im Auge behalten
- Ende 2028?!
 - Wohl kaum ;-)

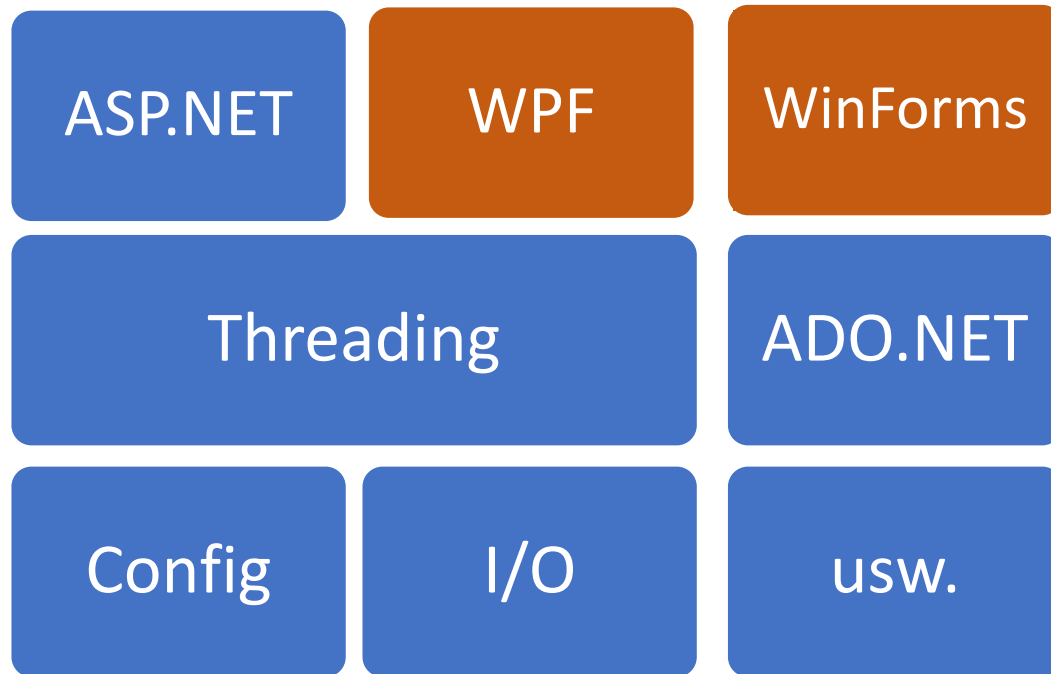


Einführung

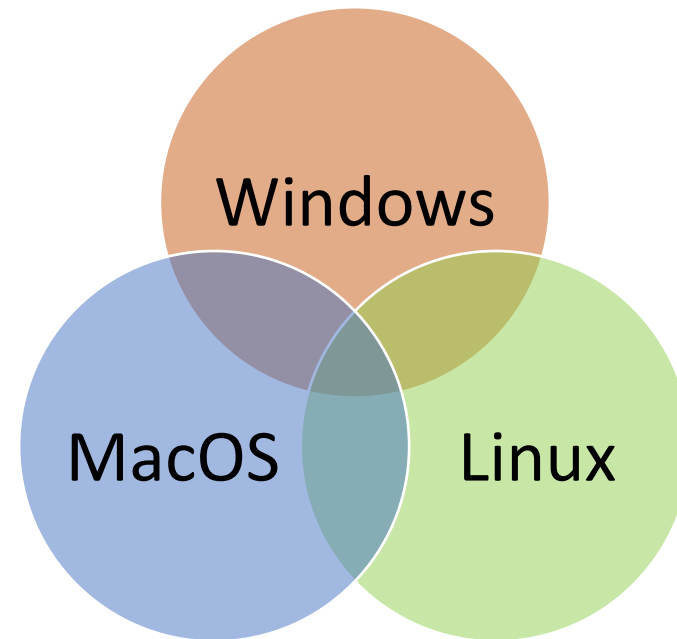


.NET Core-Aufbau

Modular

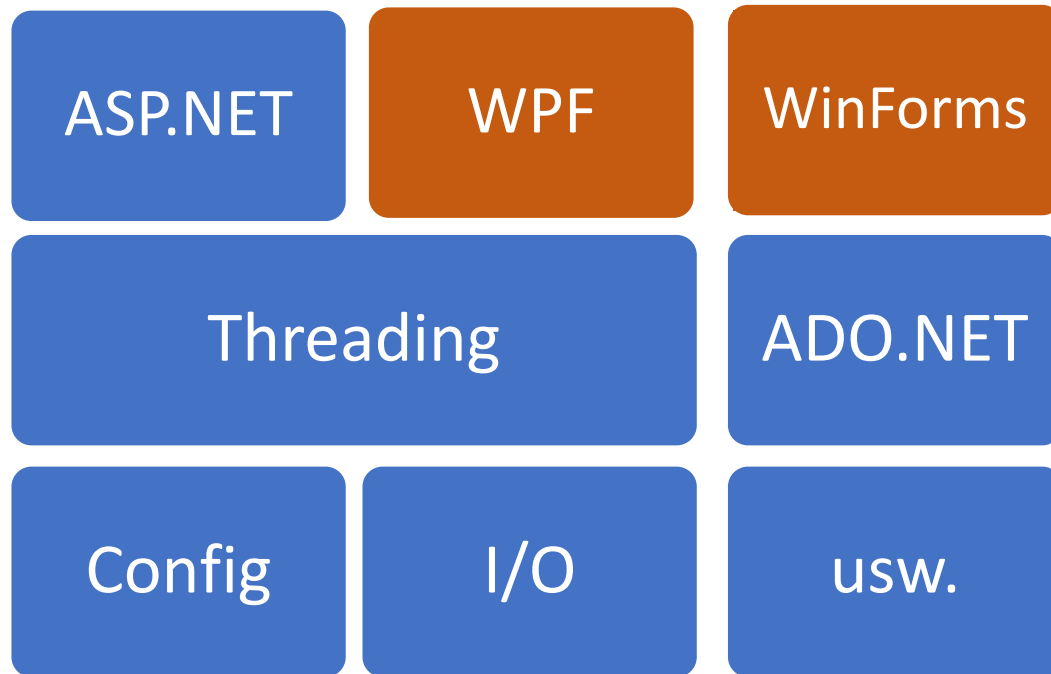


Multi Plattform

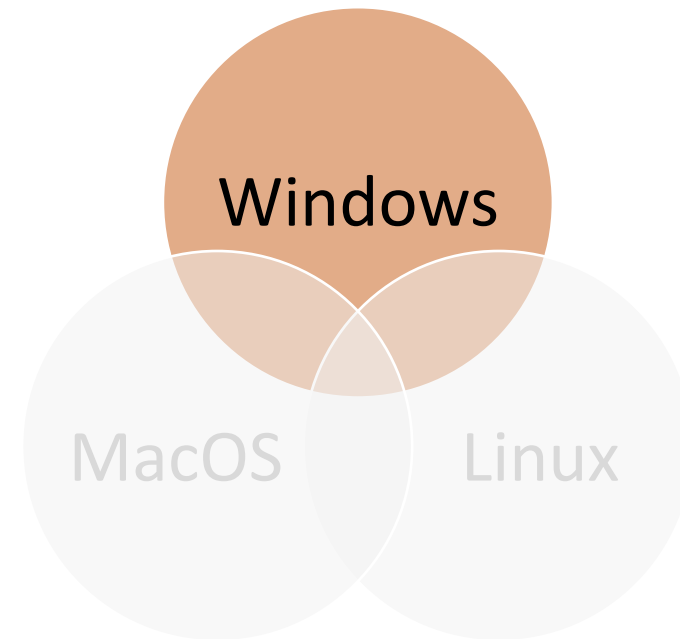


WPF unter .NET Core-Aufbau

Modular



Multi Plattform





WPF unter .NET Core

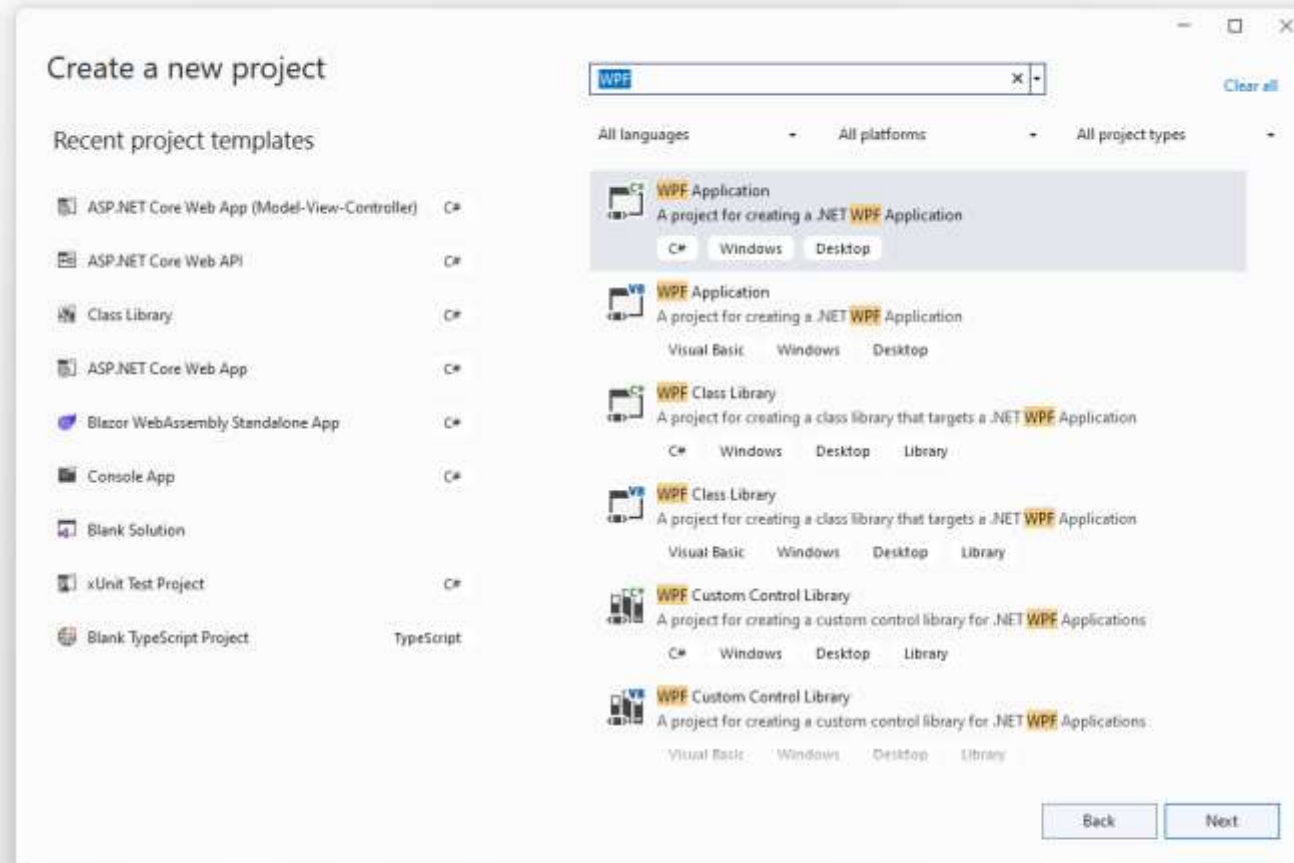
WPF unter .NET Core

- Ab .NET Core 3.0
- Visual Studio 2019+ (auch) mit Designer
- Nur unter Windows lauffähig
 - WPF basiert auf DirectX!
- Gleiches gilt übrigens auch für WinForms
 - Basiert allerdings auf GDI+

Vieles gleich, manches unterschiedlich

- XAML & Code (behind) sind identisch
- Unterschied .NET Framework vs .NET Projekten
 - keine App.Config
 - Projektdatei in SDK-Style

Visual Studio 2022+



 Demo 

.NET Core CLI

```
C:\>dotnet new
```

```
x64_x86 Cross Tools Comm... x + v
C:\Program Files\Microsoft Visual Studio\2022\Professional>dotnet new
The 'dotnet new' command creates a .NET project based on a template.

Common templates are:
Template Name      Short Name  Language  Tags
-----
ASP.NET Core Web App  webapp,razor  [C#]      Web/MVC/Razor Pages
Blazor Server App    blazorserver  [C#]      Web/Blazor
Class Library        classlib     [C#],F#,VB  Common/Library
Console App          console      [C#],F#,VB  Common/Console
Windows Forms App    winforms     [C#],VB     Common/WinForms
WPF Application      wpf          [C#],VB     Common/WPF

An example would be:
  dotnet new console

Display template options with:
  dotnet new console -h
Display all installed templates with:
  dotnet new list
Display templates available on NuGet.org with:
  dotnet new search web

C:\Program Files\Microsoft Visual Studio\2022\Professional>
```




Demo





Windows Compatibility Pack

Windows Compatibility Pack

- Code Pages
- CodeDom
- Configuration
- Directory Services
- Drawing
- ODBC
- Permissions
- Ports
- Windows Access Control Lists (ACL)
- Windows Communication Foundation (WCF)
- Windows Cryptography
- Windows EventLog
- Windows Management Instrumentation (WMI)
- Windows Performance Counters
- Windows Registry
- Windows Runtime Caching
- Windows Services



<https://docs.microsoft.com/en-us/dotnet/core/porting/windows-compat-pack>



Migration



Migration zu .NET Core

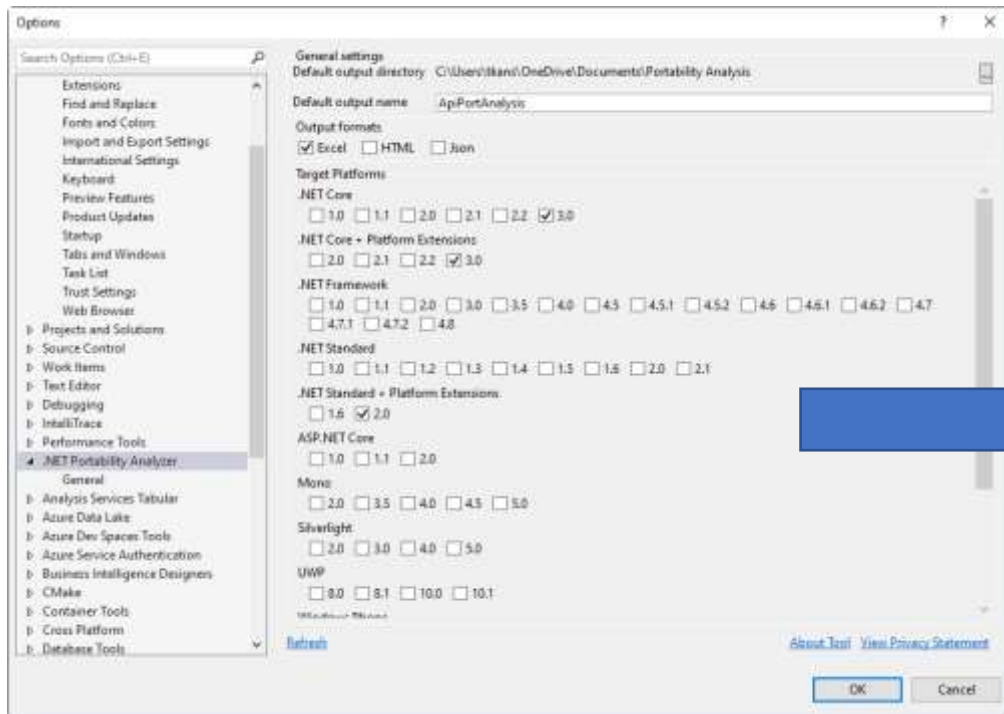
1. Aufwand abschätzen mit Analyzers
 - API Portability Analyzer Tool
 - NET Core 3.0 Desktop Api Analyzer
2. Dritthersteller (UI-)Komponenten prüfen
3. Projekte auf .NET Framework 4.6.2+ migrieren
4. Testportierung
5. Testdeployment
6. ...



<https://github.com/Microsoft/dotnet-apiport/>

.NET Portability Analyzer

Visual Studio Plug-in für 2015/ 2017/ 2019/ 2021



Submission Id	Description	Targets	.NET Core + Platform Extensions	.NET Core	.NET Framework	.NET Standard + Platform Extensions
6d640c62-a8f4-41c1-bd64-9051f6323783	dotnetconsulting.DotNetCoreLibrary	.NET Framework, Version=v4.7.2	100	78.07	78.07	100



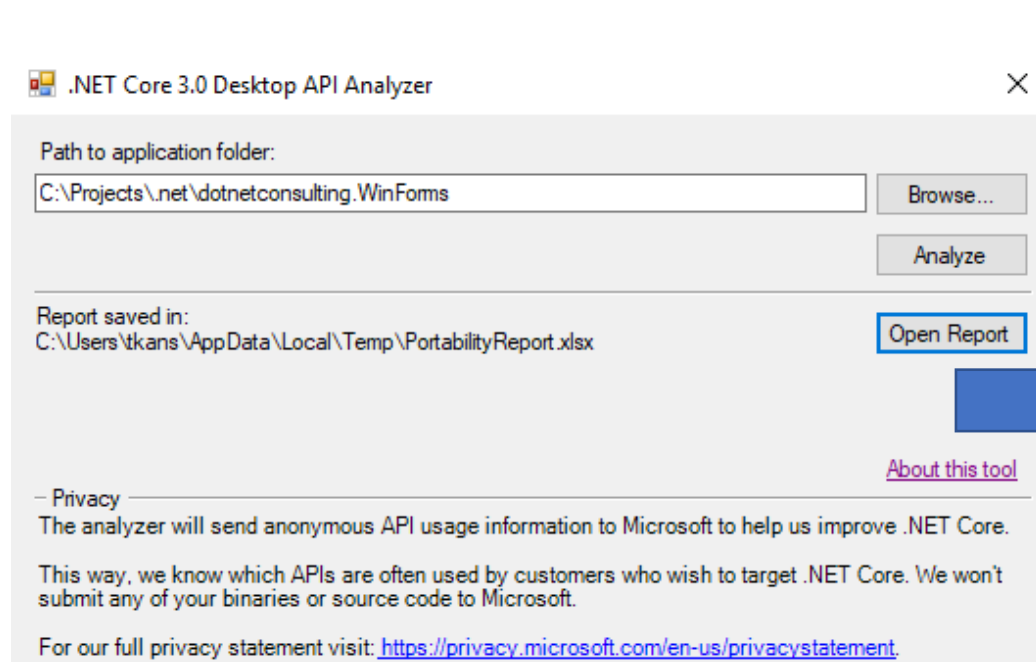
<https://docs.microsoft.com/en-us/dotnet/standard/analyzers/portability-analyzer>



Demo



.NET Desktop Api Analyzer



The screenshot shows an Excel spreadsheet with a table of API usage data. A blue arrow points from the "Open Report" button in the application window to this spreadsheet. The table has columns for Submission Id, Description, Targets, and a score. The data is as follows:

Submission Id	Description	Targets	Score
c3952345-4d95-41e9-8f3c-e8344ce240d8		.NET Core	
Header for assembly name entries			Target Framework
			.NET Core
6	dotnetconsulting.DotNetCoreLibrary		100
7	ef, Version=2.1.4.0, Culture=neutral, PublicKeyTo .NETCoreApp,Version=v2.0		100
8	ef, Version=2.1.4.0, Culture=neutral, PublicKeyTo .NETFramework,Version=v4.6.1		98.65
9	Microsoft.CSharp, Version=4.0.0.0, Culture=neut:		100
10	Microsoft.CSharp, Version=4.0.2.0, Culture=neut:		100
11	Microsoft.CSharp, Version=4.0.4.0, Culture=neut:		100
12	Microsoft.EntityFrameworkCore	.NETStandard,Version=v2.0	100
13	Microsoft.EntityFrameworkCore.Abstractions	.NETStandard,Version=v2.0	100
14	Microsoft.EntityFrameworkCore.Analyzers	.NETStandard,Version=v1.3	51.01
15	Microsoft.EntityFrameworkCore.Design, Version: .NETStandard,Version=v2.0		100
16	Microsoft.EntityFrameworkCore.Design, Version: .NETFramework,Version=v4.6.1		98.66
17	Microsoft.EntityFrameworkCore.Relational	.NETStandard,Version=v2.0	100
18	Microsoft.Extensions.Caching.Abstractions	.NETStandard,Version=v2.0	100



<https://devblogs.microsoft.com/dotnet/are-your-windows-forms-and-wpf-applications-ready-for-net-core-3-0>



Demo



Mit Assistent & viel Arbeit

GUI Migration

- Erstellung neuer Projekte in der Solution mit SDK-Projektformaten
 - WPF/ WinForm => .NET Core
 - Libraries => .NET Core oder .NET Standard
- Verlinken der ursprünglichen Quelldateien
 - Somit eine Quelldatei in mehreren Projekten
- (Nuget-)Package-Referenzen anpassen
- Namespace- und Assemblyname anpassen
- Erstellung der Assembly-Info unterdrücken
- Compiler-Fehler beheben
 - Windows Compatibility Pack?
- Try-Convert? Upgrade-Assistant? Nicht bei realen Live-Projekten...



<https://github.com/dotnet/try-convert>

<https://docs.microsoft.com/en-us/dotnet/core/porting/upgrade-assistant-wpf-framework>

SDK-Projektdatei

```
<Project Sdk="Microsoft.NET.Sdk.WindowsDesktop">
  <PropertyGroup>
    ...
    <GenerateAssemblyInfo>false</GenerateAssemblyInfo>
    <!-- Namespace- und Assemblyname anpassen -->
    <RootNamespace>WpfGui</RootNamespace>
    <AssemblyName>WpfGui</AssemblyName>
  </PropertyGroup>

  <ItemGroup>
    <!-- Ggf. Code Dateien (jedoch nicht die der der XAML-Dateien) -->
    <Compile Include="..\WpfGui/**/*.*.cs"
             Exclude="..\WpfGui/**/*.*.xaml.cs" />
    <!-- Ggf. Ressourcen -->
    <EmbeddedResource Include="..\WpfGui/**/*.*.resx" />
  </ItemGroup>
```

glob-Pattern



<https://docs.microsoft.com/en-us/dotnet/core/tools/csproj>

Und Blender: Behaviors, etc.

- Nuget: Expression.Blend.Sdk => Microsoft.Xaml.Behaviors.Wpf
- C# <http://schemas.microsoft.com/xaml/behaviors> einfügen
 - <http://schemas.microsoft.com/expression/2010/interactivity>
 - <http://schemas.microsoft.com/expression/2010/interactions>
- XAML: Microsoft.Xaml.Behaviors einfügen
 - Microsoft.Xaml.Interactivity
 - Microsoft.Xaml.Interactions



 Demo 

Test Migration

- Update NuGet dependencies
 - Standard oder .NET Core
- Update Project SDK-style Format
 - Nuget-Packages
(*packages.config => <PackageReference>-Elemente*)
- Fix build issues
- Runtime testing

A lush green landscape featuring a pond in the foreground with several flamingos. The background is filled with dense tropical vegetation, including various trees and plants. A dark blue horizontal band with a repeating flamingo pattern is overlaid across the middle of the image, containing the word "Deployment" in white text.

Deployment

Deployment

- Visual Studio 2019: Publish
 - Framework-dependent Deployment (FDD)
 - Self-contained Deployment (SCD)
 - ClickOnce Deployment
- .NET Core CLI

Visual Studio 2019+: Publish

- FDD
 - Project -> Publish
- SCD
 - Project -> Publish

```
== *.csproj-Daten ==  
<PropertyGroup>  
  <RuntimeIdentifiers>win10-x64;osx.10.11-x64;linux-x64</RuntimeIdentifiers>  
</PropertyGroup>
```



<https://docs.microsoft.com/en-us/dotnet/core/deploying/deploy-with-vs>



Demo



MSIX

MSIX ist ein Paketformat, das auf Sicherheit, Schutz und Zuverlässigkeit ausgelegt ist und auf einer Kombination aus .msi-, .appx-, App-V- und ClickOnce-Installationstechnologien basiert.


Pro	Contra
Für alle Anwendungstypen (wenn Core 3.0+)	Windows 10+
Uri Launching	Sandbox Apps
Sandbox Apps	





<https://docs.microsoft.com/de-de/windows/msix/>

MSIX

Windows (3)

 **.NET desktop development**
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.

 **Desktop development with C++**
Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.



 **Universal Windows Platform development**
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Add a new project

Recent project templates
A list of your recently accessed templates will be displayed here.

Search:

Language: Platform: Project type:

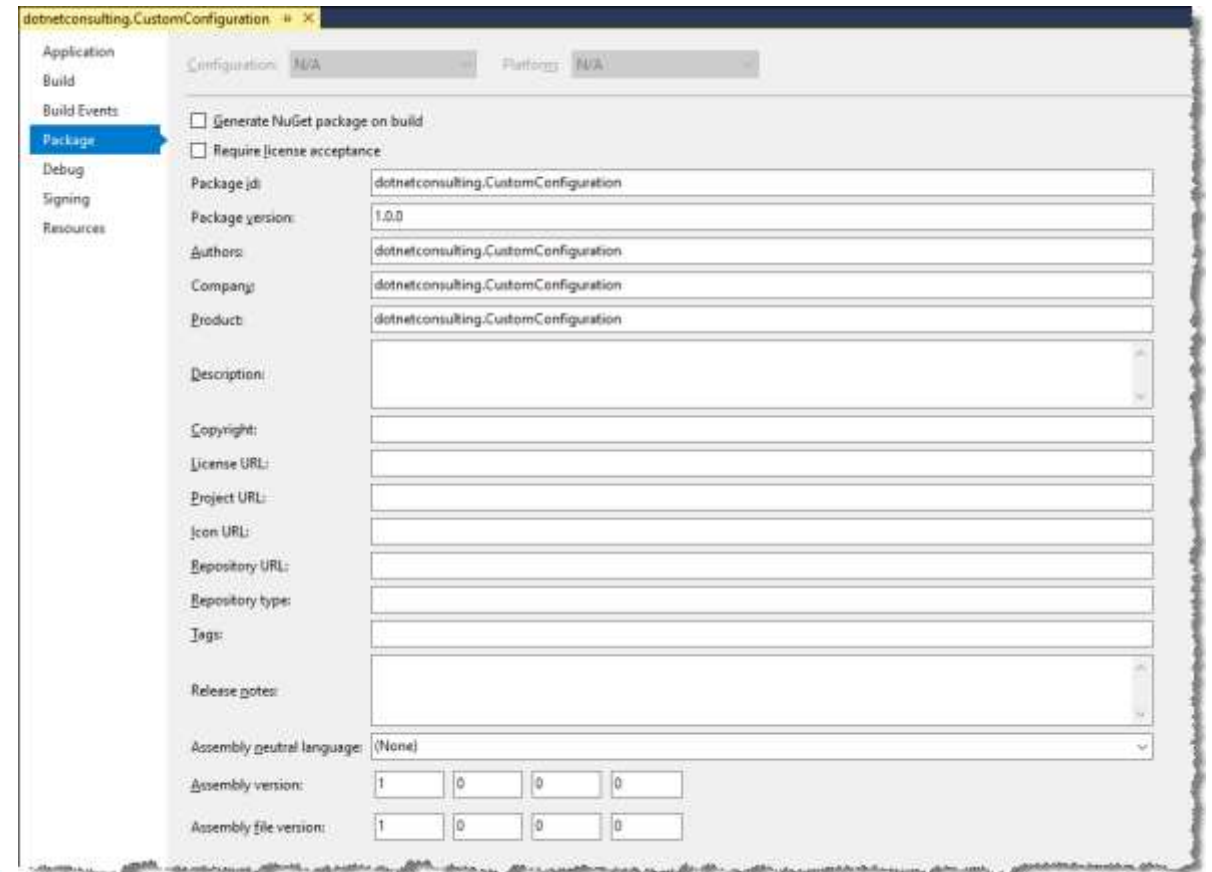
-  **Optional Code Package (Universal Windows)**
A project for creating an optional package that can contain executable code to be called from a main package.
C# Windows Xbox UWP
-  **Windows Application Packaging Project**
A project that creates packages containing Windows applications for side-loading or distribution via the Microsoft Store.
C# Windows UWP Desktop



<https://montemagno.com/distributing-a-net-core-3-wpf-and-winforms-app-with-msix/>

Visual Studio 2019+: Pack

- Erstellt ein NuGet-Paket
 - `nuget.exe` aus VS heraus
- Details:
 - Projekt -> Properties -> Package





Demo



.NET Core CLI

Framework-dependent Deployment

```
dotnet publish -c Release
```

Self-contained Deployment

```
dotnet publish -c Release -r win10-x64  
dotnet publish -c Release -r ubuntu.16.10-x64  
dotnet publish -c Release -r osx.10.11-x64
```

 Demo 

Fragen? Jetzt oder später!

Kontakt

 **E-Mail**
tkansy@dotnetconsulting.eu

 **Telefon**
[+49 \(0\) 6187 / 2009090](tel:+49(0)61872009090)

 **Microsoft Teams**
[Meet now](#)

 **LinkedIn**
[Link me](#)

 **XING**
[Xing me](#)

 **X (Twitter)**
[@tkansy](#)



www.dotnetconsulting.eu

SQL Server meets .NET (Core)- professionally!



Ich berate, coache und trainiere im Bereich Entwicklung von .NET (Core) Anwendungen mit Microsoft SQL Server- mit Allem, was dazu gehört- und was man vielleicht weglassen sollte.