

.NET upgrade

TOMÁŠ HAVETTA – MCT

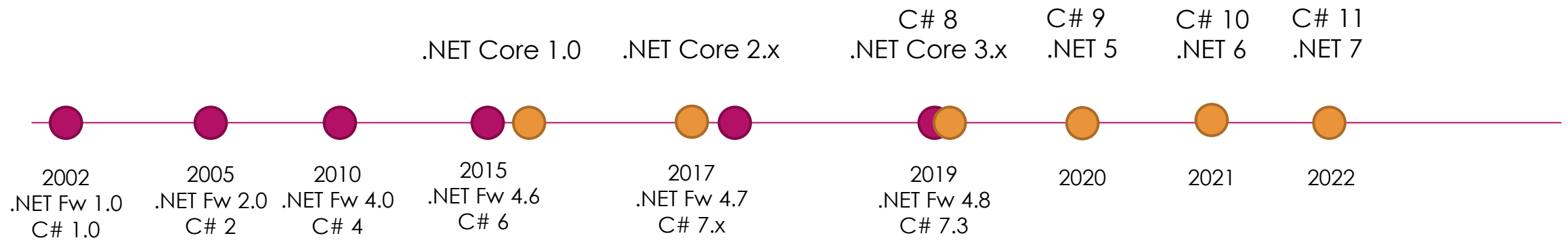
Obsah

- ▶ .NET platformy
- ▶ Migrace
- ▶ Kompilátor ROSLYN
- ▶ .NET services
- ▶ C# novinky od verze 8

.NET platformy

- ▶ Přehled platformy .NET
- ▶ LTS verze
- ▶ .NET Standard
- ▶ Rozdíly oproti .NET Framework, nepodporované technologie, náhradní/nová funkcionality
- ▶ Nová struktura projektu

Historie



Platformy (do roku 2019)

.NET Framework

Windows only
1.0 – 4.8
Desktop, Web,
Console, Services

.NET Core

Windows, Linux,
MacOS
Web, Console,
Desktop (Win only)
Services

XAMARIN

iOS + Android
Mobilní app

.NET Standard

Library only project

Současnost

- ▶ Každý rok nová verze, v listopadu
- ▶ Každý lichý rok je verze LTS = 3 roky podpora
- ▶ Podpora non LTS končí po vydání nové verze
- ▶ Aktuální .NET 7 není LTS

.NET Standard

- ▶ Původně určen pro tvorbu knihoven pro libovolnou platformu
- ▶ Definuje pouze API, nemá „runtime“
- ▶ Poslední verze je 2.1
- ▶ .NET 5 spojil kód .NET Core a .NET Standard do jedné platformy
- ▶ **.NET Standard 2.0 je poslední platforma podporující .NET Framework**

Co zmizelo přechodem na .NET - 1

- ▶ Application Domains
 - ▶ Třída AppDomains existuje, ale proces má jen jednu aplikační doménu
 - ▶ Některé metody fungují, některé nedělají nic a některé vrátí výjimku
- ▶ .NET Remoting
 - ▶ Technologie z dob .NET Framework 1.0
- ▶ Code access security (CAS)
 - ▶ Používalo se do .NET Framework 4.0
 - ▶ Nastavování práv per assembly

Co zmizelo přechodem na .NET - 2

- ▶ System.EnterpriseServices
 - ▶ Historická technologie navázaná na OS Windows, komunikace mezi procesy
- ▶ WCF Server
 - ▶ WCF je kódem extrémně svázané s OS Windows
 - ▶ Existuje náhrada **CoreWCF**, ale není 100% kompatibilní a je to verze 1.x
- ▶ Workflow Foundation
 - ▶ Mrtvola od .NET Framework 4.0
- ▶ XSLT Script block
 - ▶ MS rozšíření XSLT

Stěžejní výhody .NET x

- ▶ Rychlost
- ▶ Rychlost
- ▶ Rychlost
- ▶ Multiplatformovost

.NET x project

- ▶ Nová forma a pravidla pro .NET project file (*.csproj)
- ▶ XML
- ▶ Neobsahuje seznam souborů. Ty jsou dány umístěním v souborovém systému
- ▶ Mnoho voleb se ovládá přímou editací, protože UI VS 20xx nestíhá
- ▶ <https://learn.microsoft.com/en-us/dotnet/standard/frameworks>

Vývojové prostředí

- ▶ Visual Studio 20xx
 - ▶ Community, Professional, Enterprise
 - ▶ <https://visualstudio.microsoft.com/cs/vs/compare/>
- ▶ Visual Studio Code
 - ▶ Pro nadšence CLI
 - ▶ Výborné pro editaci non C#

Migrace

- ▶ Scénáře migrace projektu z .NET Framework na .NET (Core)
- ▶ Pomocné a (polo)automatické nástroje

Možné scénáře

- ▶ Nový projekt + přesun zdrojových souborů + opětovné nastavení závislostí
 - ▶ Vhodné pro jednoduché projekty
 - ▶ Vše pod Vaší kontrolou
- ▶ Migrace pomocí nástrojů
 - ▶ Použitelné pro velké projekty s mnoha vazbami
 - ▶ Vyžaduje trpělivost a zkušenost v CLI
 - ▶ Negarantuje 100% úspěch

Příprava

- ▶ Upgrade projektu na .NET Framework 4.8
- ▶ Aktualizace a kontrola referencovaných knihoven
- ▶ Analýza assembly zda obsahuje nekompatibilní API
 - ▶ VS 2017/2019 - .NET Portability Analyzer
 - ▶ VS 2022 - Binary Analysis in Upgrade Assistant (experimental)
https://github.com/dotnet/upgrade-assistant/blob/6f249b66390413e3496fbd619f2cff488259b51e/docs/binary_analysis.md

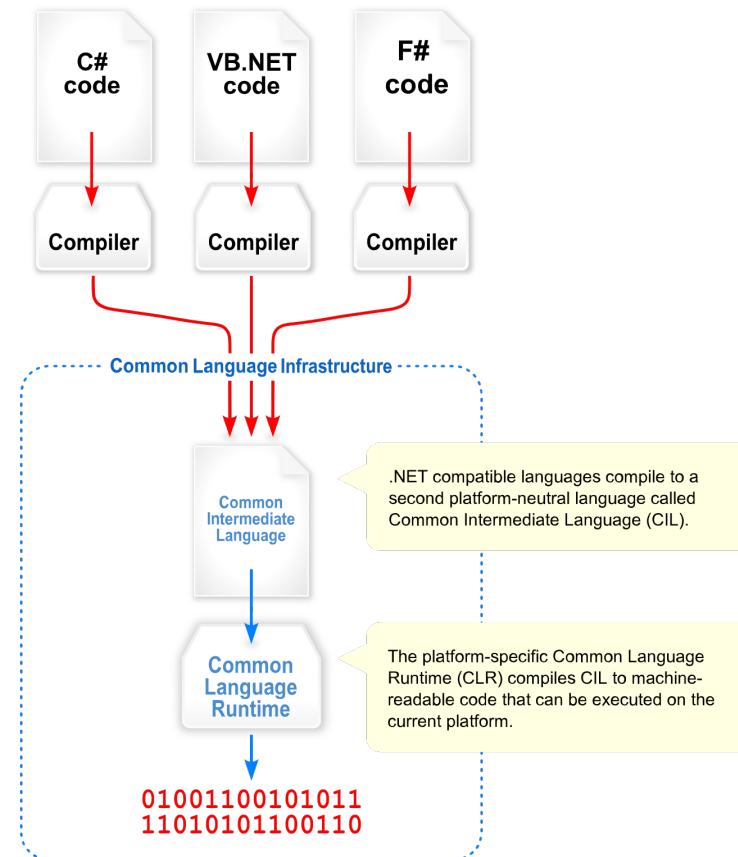
EF 6 a EF Core

- ▶ .NET x podporuje obě verze! Migrace tak není nutností
- ▶ EF Core NENÍ pokračování EF 6
 - ▶ <https://learn.microsoft.com/en-us/ef/efcore-and-ef6/>
 - ▶ EF 6.4 se už jen udržuje
- ▶ ***Just because your application compiles, does not mean it is successfully ported to EF Core.*** You will need to test all areas of your application to ensure that none of the behavior changes have adversely impacted your application. – oficiální dokumentace k migraci EF 6 na EF Core

Kompilátor ROSLYN

- ▶ Targeting knihoven, SDK, verze runtime
- ▶ Multiplatformnost (x-plat)
- ▶ Distribuce aplikací, self-contained deployment, roll forward
- ▶ AOT, PGO, R2R a související technologie

CLI – Common Language Infrastructure



Proč ROSLYN

- ▶ ROSLYN není black box
- ▶ Nabízí API pro vstup nejen do procesu kompilace
- ▶ Integrace s VS nástroji
- ▶ Standardizace kódu
- ▶ Oprava kódu
- ▶ Generování kódu

.NET x komponenty

- ▶ Konfigurační soubory
- ▶ Změny v runtimu a GC, nové možnosti programování v C# pro optimalizovaný kod
- ▶ Event counters
- ▶ Reflection, source generators
- ▶ Structured logging
- ▶ Přehled integrovaného IoC/DI

Konfigurace - IConfiguration

- ▶ JSON
- ▶ XML
- ▶ INI
- ▶ Env variables
- ▶ Cmd line
- ▶ Azure
- ▶ Kdekoli pomocí vlastních komponent

Logování - ILogger

- ▶ Vlastní nic moc
- ▶ Serilog
- ▶ Net4Log
- ▶ Jakékoliv vlastní

DI/loC - IServiceProvider

- ▶ DI – omezit závislosti
- ▶ loC – abstraktní typ nesmí být závislý na konkrétním typu
- ▶ .NET implementace DI není jedinou možností

C# od verze 8

- ▶ .NET a C#
- ▶ C# 8 break changes
- ▶ C# 9 Records
- ▶ C# 10 Record Struct
- ▶ C# 11 co ještě ???

Vazba .NET a C#

.NET verze	C# verze
.NET Framework 4.8	7.3
.NET Core 3.1	8
.NET 5	9
.NET 6	10
.NET 7	11
.NET Standard 2.0	7.3
.NET Standard 2.1	8

C# 8 – breaking changes

- ▶ Default Interface
- ▶ Nullable referenčné typy

Default Interface

- ▶ Umožňuje zápis kódu do deklarace interface
- ▶ Třída implementující interface nemusí definovat metodu interface, která má kód!
- ▶ Je možné přidat metody do interface bez nutnosti opravit všechny třídy, které interface implementují

Nullable referenční typy

- ▶ Povolení null pro referenční typy byla největší chyba návrhu C# v roce 2000
- ▶ Top výjimka .NET aplikací – `NullReferenceException`
- ▶ Doporučuji začít striktně používat!

Další vychytávky C# 8

- ▶ Async stream
- ▶ Ranges
- ▶ switch expression

C# 9 - Record

- ▶ Označení immutable class
- ▶ Předpokládá správnou deklaraci !!!
- ▶ Class který se chová jako hodnotový typ

Další zajímavosti C# 9

- ▶ Top-level statement
- ▶ Init setter
- ▶ Pattern Matching nekončí v novinkách ...
- ▶ new bez typu
- ▶ Partial methods – uvolnění restrikcí

C#10 – Record Struct

- ▶ Record struct => aby strukturám nebylo líto
- ▶ global using => pro líné
- ▶ File scope namespace declaration
- ▶ Custom Interpolated String Handler => pro ty co je baví kódovat
- ▶ K lambda výrazům lze přidávat atributy

C#11 – co bychom tam tak ještě dali???

- ▶ Generické atributy
- ▶ Matematické výpočty
- ▶ Required members
- ▶ Numeric IntPtr
- ▶ File local scope => file access modifier
- ▶ Nový operátor >>>
- ▶ ...