

.NET upgrade

TOMÁŠ HAVETTA – MCT

1

Obsah

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

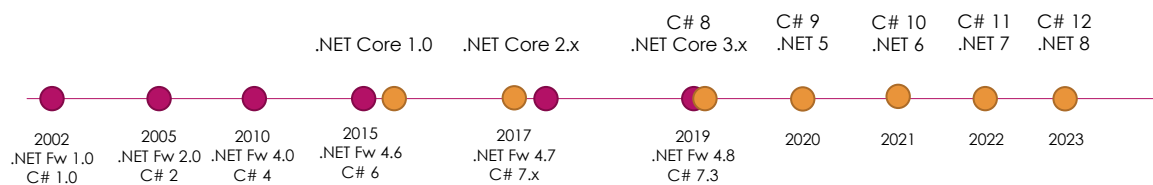
2

.NET platformy

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

3

Historie



4

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

5

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 8 je LTS

6

.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**

7

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

8

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

9

Stěžejní výhody .NET x

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

10

.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>

11

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#

12

Migrace

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

13

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

14

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

15

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

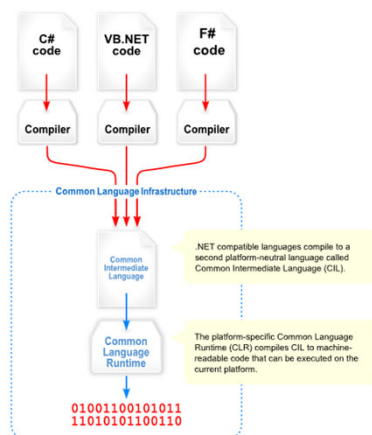
16

Kompilátor ROSLYN

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

17

CLI – Common Language Infrastructure



By Deviousasti - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=88880454>

18

Proč ROSLYN

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

19

.NET x komponenty

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

20

Konfigurace - IConfiguration

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

21

Logování - ILogger

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

22

DI/loC - IServiceProvider

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

23

C# od verze 8

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

24

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 8	12
.NET Standard 2.0	7.3
.NET Standard 2.1	8

25

C# 8 – breaking changes

- ▶ Default Interface
- ▶ Nullable referenčné typy

26

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

27

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!

28

Další vychytávky C# 8

- ▶ Async stream
- ▶ Ranges
- ▶ switch expression

29

C# 9 - Record

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

30

Další zajímavosti C# 9

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

31

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

32

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

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

33

C# 12 - přidáváme

- ▶ Primary constructors
- ▶ Collection expressions a operátor .. pro kolekce
- ▶ ref readonly parameters (řeší že in mění podpis metody)
- ▶ Default lambda parameters + params parameter
- ▶ Inline array => array jako struktura na zásobníku (už nic neplatí, C se blíží)

34

Knihovny

- ▶ MediatR – Mediator pattern
- ▶ FluentValidation – formálně validuje
- ▶ Serilog – super logger .NET
- ▶ Refit – type-safe REST client
- ▶ Polly – jak se vzpamátovat z chyby
- ▶ Moq – unit testy a MOQ
- ▶ EF Core / Dapper – vyberte si ORM
- ▶ StackExchange.Redis – in-memory DB
- ▶ MassTransit – tvorba distrib. app
- ▶ Mapster – mapování mezi třídami
- ▶ SonarAnalyzer.Csharp – VS extenze
- ▶ Autofac – DI jinak

35

Děkuji za pozornost - hodnocení

tomas@havetta.cz

36