

# Sourcelink - code on demand

Patrik Švikruha

# Obsah

- Pdb
  - Microsoft || Windows PDB
  - Portable PDB
- Build konfigurácia
  - Debug vs Release
- Možnosti debugovania dll
  - Symbol server
  - Symbol package
  - Dekompilácia
- SourceLink
  - Demo

Windows PDB || Microsoft PDB ||  
PDB || Portable PDB || MNB




# PDB == Microsoft PDB

- *program database* (.pdb) - binárna forma
- Vznikol pred viac ako 25 rokmi – v.7
- Vytvára ho linker || Roslyn (managed)
- Slúži na oddelenie debug a pdb info od funkčného kódu (dll/exe)
  - Určitá ochrana proti reverse engineering
  - (Ne)znižuje sa tým veľkosť zdrojáku - dll

# PDB == Microsoft PDB

- Ukladá **debug info** a **pdb info** o danej dll
  - Obsahuje cesty k zdrojovým súborom
  - Obsahuje názvy a informácie o typoch
  - Obsahuje public symbols
  - Mapuje inštrukcie na riadky kódu
- Debbuger ho defaultne hľadá pri dll (exe) alebo na Symbol serveroch

# PDB == Microsoft PDB

- Dokumentácia? 
  - Source code is the ultimate documentation :-) ([link](#))
- Proprietárny formát
- Navrhnuté pre unmanaged kód (C,C++)
- Podpora pre .NET je „iba“ pridaná
  - Resp debugger ([link](#))
- Windows only

# Portable PDB != (PDB || Microsoft PDB)

- .NET Core je open source
  - Microsoft PDB nemohli použiť kvôli licenciam
- Navrhnuté iba pre managed kód .NET
  - Portable PDB je navrhnutý nanovo
  - Menšia veľkosť (1KB vs. 12KB)
- Cross-platform && open-source && documented
  - Dokumentácia ([link](#)), Špecifikácia ([link](#)), Repo ([link](#))
- Spravuje to .NET foundation

# Portable PDB != (PDB || Microsoft PDB)

- Nevýhody z dokumentácie:
  - Sú určené iba pre .NET svet ( nie pre C/C++)
  - Staršie verzie VS2015 update 2
  - Aplikácie ktoré targetujú <=.NET framework 4.7.1
  - Code Contracts
  - C# style analyzátor FxCop
  - Symbol server
  - Dekompilátory ako .Net reflector a ildasm
  - [MS DIA](#)(debug interface access) - WinDBG



# Build konfigurácia - Debug

- Optimalizácia kódu je vypnutá (Roslyn + JIT)
- GC nekolektuje tak často
- IL nop instruction
  - Napr. breakpoint na poslednej curly bracket
- Full debug info (pdb + debug info)

# Build konfigurácia - Release

- Optimalizácia kódu (Method/var inlining)
- PDB only (bez debug info)
- Optimalizácia zapnutá na úrovni Roslyn a aj JIT kompilátora
- Článok ([link](#))

# Možnosti debugovania dll



Remy Sharp • @rem • Fronteers 2015

# Možnosti debugovania dll

- Symbols (PDBs)
  - Súbor bez ktorého sa nedá debuggovať
  - Sú (boli) rovnaké pre managed a unmanaged kód
- Zdrojové kódy
  - Dekompilácia
  - Source server
  - Repozitár (git, TFS, ~~SVN~~)

# Debugging nuget packages

- Problémy pri ladení aplikácii s použitím balíčkov.
- Čo keby sa dali nuget balíčky debuggovať?
  - Symbol Server
  - Symbol package
  - Dekompilation e.c. dotpeek. .NET Reflector, etc.
  - Source link

# Symbols server

- V podstate sa jedná o „server“ voči ktorému sa debugger dotázal o potrebné PDB
- Výhoda je v tom že mám všetky PDB na jednom mieste – do VS/WinDbg stačí pridať 1 uri na server.
- Má ho napr aj ~~VSTS~~ Azure DevOps
  - V CI (Pipelines) je možnosť pridať build step
- Používa sa v kombinácii s Source server

# Symbols package

- `nuget pack XX.csproj -Symbols`
- `dotnet pack --include-symbols --include-source`
- `{identifrier}.symbols.nupkg`
  - Obsahuje dll, pdb
  - Môže obsahovať XMLDoc, sourceFiles
  - Vytvoriť symbol package je jednoduché
- Nevýhoda vo veľkosti balíčku

# Dekompilácia

- Nie je to originálny zdroják
- Pdb si tiež musím vygenerovať
  - Podstatný je rovnaký GUID ako v dll-ke
  - Dll je zbuildená (väčšinou) v Release konfigurácii
- Nemôžem meniť hodnoty variables..



# Source link

- „SourceLink enables a great source debugging experience for your users, by adding source control metadata to your built assets“
- Code on demand pre debugger z repozitára
  - stačí mi pdb kde je link a commitId do repozitára
- Podpora GitHub, GitHub Enterprise, Azure DevOps (git || TFS), GitLab, Bitbucket.org

# Source link

- „SourceLink enables a great source debugging experience for your users, by adding source control metadata to your built assets“
- Code on demand z repozitára – stačí mi pdb s linkom a commitId do repozitára
- Podpora GitHub, GitHub Enterprise, Azure DevOps (git || TFS), GitLab, Bitbucket.org

# Source link

- Výhoda
  - Dokážem presne debugovať zdrojový kód
  - Môžem meniť variables
  - Môžem meniť flow kódu
- Nevýhoda
  - Edit and Continue nefunguje (má to svoj dôvod)
  - Je potrebné aby vydávateľ balíčku pridal **source link nuget** do svojej knižnice
  - Ideálne aby pridaj aj „-debug“ package

# Source link = package

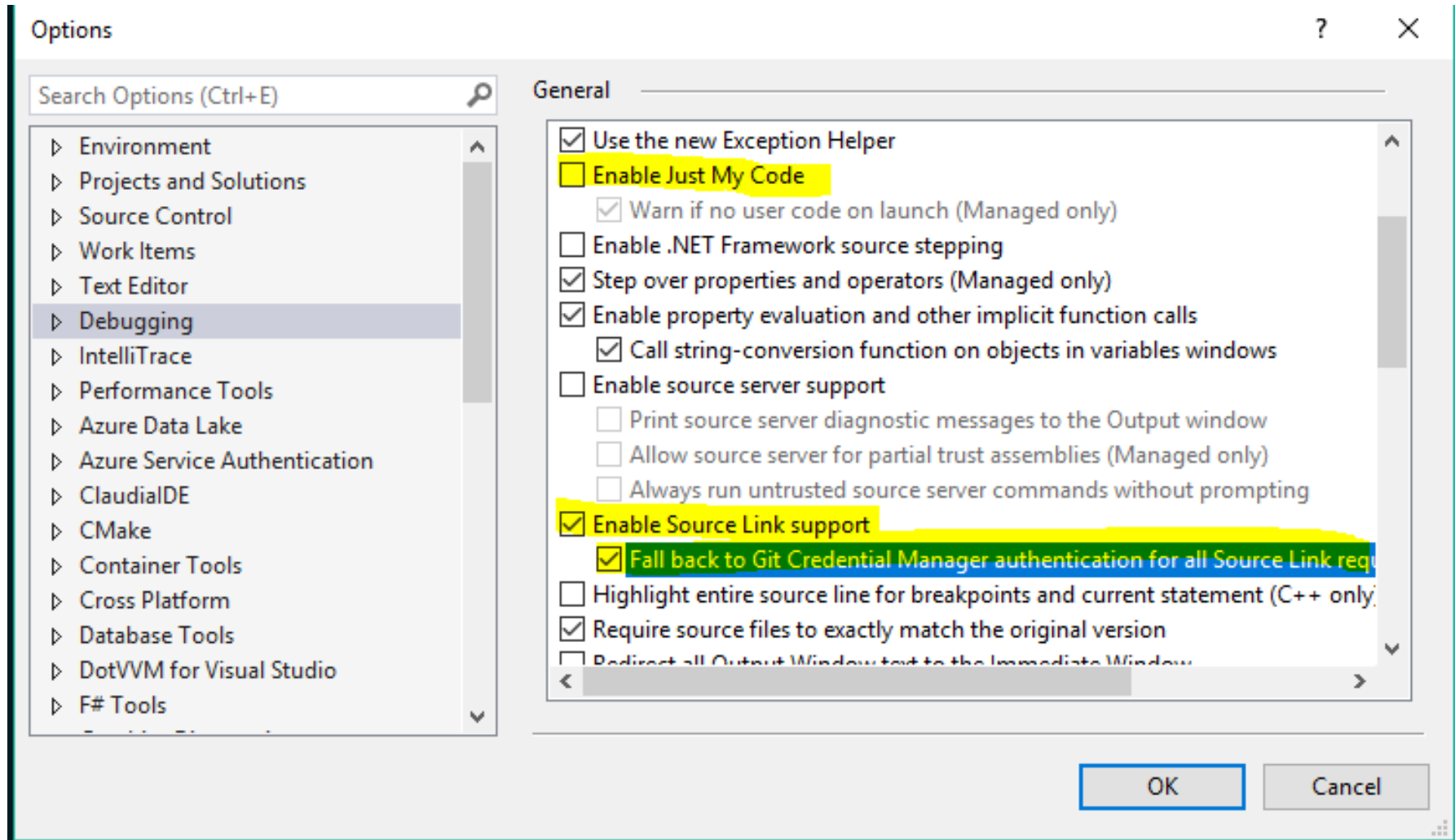
```
Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>netstandard2.0</TargetFramework>
    <!-- Optional: Publish the repository URL in the built .nupkg (in the
NuSpec <Repository> element) -->
    <PublishRepositoryUrl>true</PublishRepositoryUrl>
    <!-- Optional: Include the PDB in the built .nupkg -->
    <AllowedOutputExtensionsInPackageBuildOutputFolder>$(AllowedOutputExte
nsionsInPackageBuildOutputFolder);.pdb</AllowedOutputExtensionsInPackageBuildO
utputFolder>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.SourceLink.Vsts.Git"
Version="1.0.0-beta-63127-02" PrivateAssets="All"/>
  </ItemGroup>
</Project>
```

# Source link = konzument

- Disable „Enable just my code“
- Enable „Enable Source link support“
- Podpora pre VS

# Source link = konzument



# Source link

- DEMO
  - <https://github.com/PadreSVK/sourcelinkdemo>
- Nuget
  - NetStandard 2.0
  - NetStandardPackage
- CI(Pipeline)
  - Vytvorí 2 package (jeden so suffixom -debug)