
Hydrogen Sessions

#7 – `Hydrogen.DesignTime.Configuration`

© 2020 PRIMAVERA

Builds Configuration

O package `Primavera.Hydrogen.DesignTime.Configuration`, que é adicionado em todos os projetos, faz essencialmente 3 coisas muito importantes:

- Configura a SNK para assinar as assemblies.
- Configura os analyzers para validar regras de code analysis.
- Configura as seguintes propriedades dos projetos:
 - `CopyLocalLockFileAssemblies`
 - `GenerateAssemblyInfo`
 - `GenerateAssemblyConfigurationAttribute`

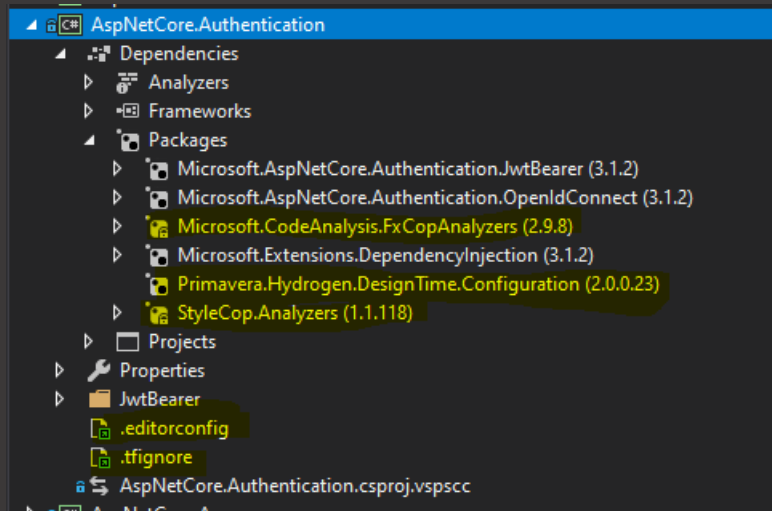
Trata-se de um package NuGet reconhecido pelo Visual Studio como uma “development dependency” (em vez de “binary dependency”).

O source code está em `$/Lithium/Core/Hydrogen.DesignTime/Mainline` (se tiverem curiosidade sobre como se constroem packages como este).

Nos projetos

Quando o package é adicionado a um projeto:

- São adicionadas as dependências para o package e para os analyzers
- São adicionados os ficheiros `.editorconfig` e `.tfignore` ao projeto

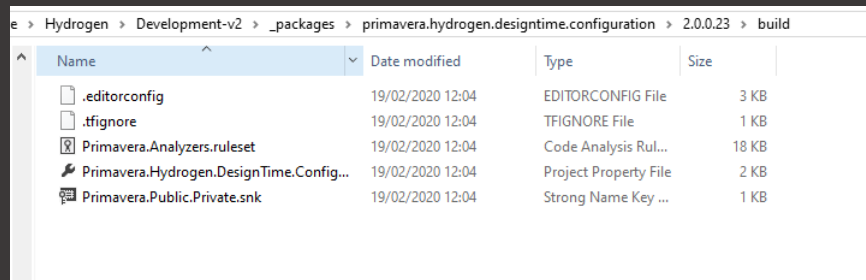


É fundamental que...

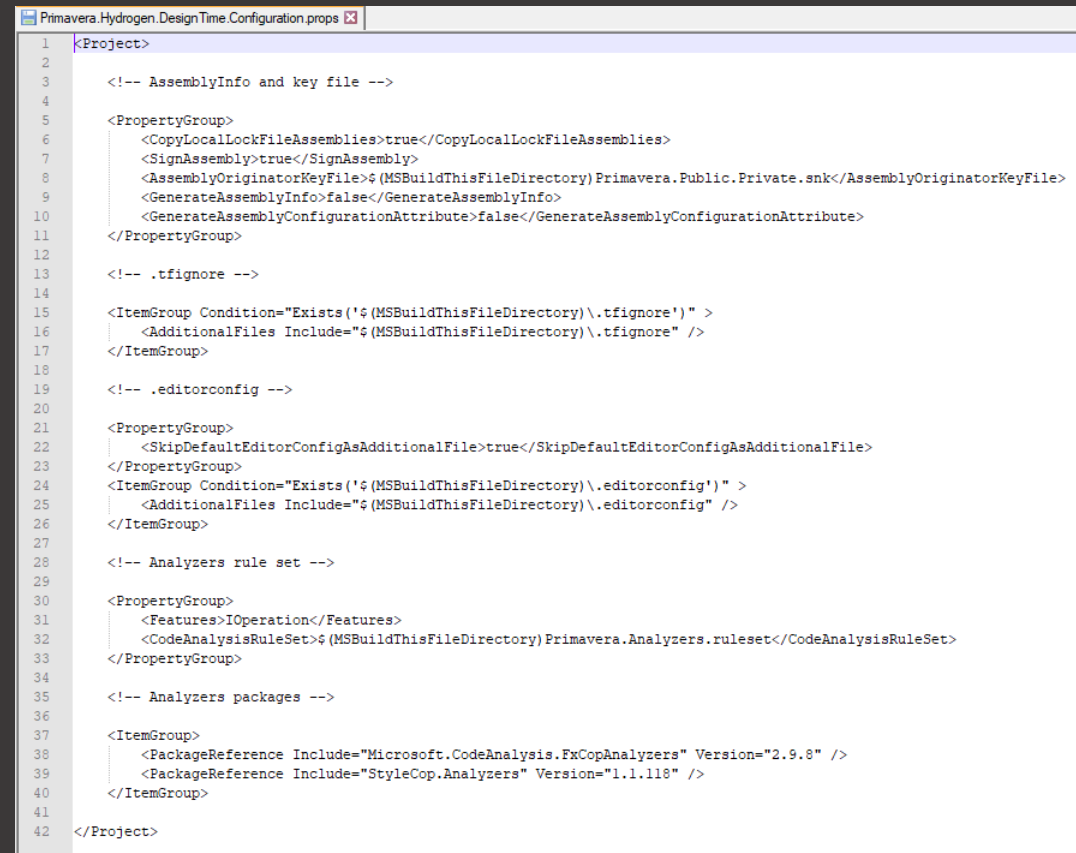
- Os packages `Microsoft.CodeAnalysis.FxCopAnalyzers` e `StyleCop.Analyzers` NUNCA sejam atualizados manualmente.
- Os dois ficheiros NUNCA sejam removidos ou modificados de nenhuma forma.
- Esses ficheiros nunca vão parar ao source control (periodicamente o Visual Studio tem bugs relacionados com estes ficheiros que parecem fazer parte do projeto mas não estão na realidade na diretoria do projeto).

Como funciona?

Se analisarem o package restaurado nos projetos, é possível perceber como que é isto funciona.



Name	Date modified	Type	Size
.editorconfig	19/02/2020 12:04	EDITORCONFIG File	3 KB
.tfignore	19/02/2020 12:04	TFIGNORE File	1 KB
Primavera.Analyzers.ruleset	19/02/2020 12:04	Code Analysis Rul...	18 KB
Primavera.Hydrogen.DesignTime.Config...	19/02/2020 12:04	Project Property File	2 KB
Primavera.Public.Private.snk	19/02/2020 12:04	Strong Name Key ...	1 KB



```
1 <Project>
2
3 <!-- AssemblyInfo and key file -->
4
5 <PropertyGroup>
6   <CopyLocalLockFileAssemblies>true</CopyLocalLockFileAssemblies>
7   <SignAssembly>true</SignAssembly>
8   <AssemblyOriginatorKeyFile>$(MSBuildThisFileDirectory)Primavera.Public.Private.snk</AssemblyOriginatorKeyFile>
9   <GenerateAssemblyInfo>>false</GenerateAssemblyInfo>
10  <GenerateAssemblyConfigurationAttribute>>false</GenerateAssemblyConfigurationAttribute>
11 </PropertyGroup>
12
13 <!-- .tfignore -->
14
15 <ItemGroup Condition="Exists('$(MSBuildThisFileDirectory)\.tfignore') " >
16   <AdditionalFiles Include="$(MSBuildThisFileDirectory)\.tfignore" />
17 </ItemGroup>
18
19 <!-- .editorconfig -->
20
21 <PropertyGroup>
22   <SkipDefaultEditorConfigAsAdditionalFile>true</SkipDefaultEditorConfigAsAdditionalFile>
23 </PropertyGroup>
24 <ItemGroup Condition="Exists('$(MSBuildThisFileDirectory)\.editorconfig') " >
25   <AdditionalFiles Include="$(MSBuildThisFileDirectory)\.editorconfig" />
26 </ItemGroup>
27
28 <!-- Analyzers rule set -->
29
30 <PropertyGroup>
31   <Features>IOperation</Features>
32   <CodeAnalysisRuleSet>$(MSBuildThisFileDirectory)Primavera.Analyzers.ruleset</CodeAnalysisRuleSet>
33 </PropertyGroup>
34
35 <!-- Analyzers packages -->
36
37 <ItemGroup>
38   <PackageReference Include="Microsoft.CodeAnalysis.FxCopAnalyzers" Version="2.9.8" />
39   <PackageReference Include="StyleCop.Analyzers" Version="1.1.118" />
40 </ItemGroup>
41
42 </Project>
```

Repare-se:

- Na configuração dos analyzers (package references).
- Na configuração do ruleset utilizado na validação do code analysis.
- Na configuração do ficheiro .editorconfig

Ruleset + .editorconfig

O Visual Studio 2019 está num processo de migração (digamos assim) de rulesets para .editorconfig.

Como o processo está longe de suportar direito cenários como o nosso – em que se pretende definir regras transversais a vários projetos e a várias equipas de desenvolvimento – temos que viver com as duas soluções. A parte mais importante das regras está ainda definida no ruleset, mas algumas já são configuradas no .editorconfig.

<https://github.com/dotnet/roslyn/issues/38782>

```
Primavera.Analyzers.ruleset  Primavera.Hydroge...nfiguration.nuspec  _readme.md  Source Control Explorer  .editorconfig  Prin
1  <?xml version="1.0" encoding="utf-8"?>
2  <RuleSet
3      Name="PRIMAVERA Analyzers Rule Set"
4      Description="This rule set contains all rules appropriate for PRIMAVERA projects."
5      ToolsVersion="10.0">
6      <IncludeAll Action="Warning" />
7      <Rules AnalyzerId="Microsoft.CodeQuality.Analyzers" RuleNamespace="Microsoft.CodeQuality.Analyzers">
8          <Rule Id="CA1000" Action="Info" />
9          <Rule Id="CA1008" Action="Info" />
10         <Rule Id="CA1010" Action="Info" />
11         <Rule Id="CA1012" Action="Warning" />
12         <Rule Id="CA1014" Action="None" />
13         <Rule Id="CA1016" Action="Error" />
14         <Rule Id="CA1017" Action="Warning" />
15         <Rule Id="CA1018" Action="Error" />
16         <Rule Id="CA1024" Action="Info" />
17         <Rule Id="CA1027" Action="Warning" />
18         <Rule Id="CA1028" Action="Error" />
19         <Rule Id="CA1030" Action="Warning" />
20         <Rule Id="CA1031" Action="Error" />
21         <Rule Id="CA1033" Action="Error" />
22         <Rule Id="CA1034" Action="Error" />
23         <Rule Id="CA1036" Action="Error" />
24         <Rule Id="CA1041" Action="Error" />
25         <Rule Id="CA1043" Action="Error" />
26         <Rule Id="CA1044" Action="Error" />
27         <Rule Id="CA1050" Action="Error" />
28         <Rule Id="CA1051" Action="Error" />
29         <Rule Id="CA1052" Action="Error" />
30         <Rule Id="CA1054" Action="Warning" />
31         <Rule Id="CA1055" Action="Warning" />
32         <Rule Id="CA1056" Action="Warning" />
33         <Rule Id="CA1060" Action="Error" />
```

```
Primavera.Analyzers.ruleset  Primavera.Hydroge...nfiguration.nuspec  _readme.md  Source Control Explorer  .editorconfig
1  #####
2  # Lithium Editor Configuration
3  # Version 0.2, Feb 2020, CA1508 severity
4  # Version 0.1, Sep 2019
5  #
6  # This file was imported by the Primavera.Hydrogen.DesignTime.Configuration package.
7  # DO NOT EDIT MANUALLY!
8  #
9  # More info:
10 # https://docs.microsoft.com/en-us/visualstudio/ide/editorconfig-code-style-settings-reference
11 #####
12
13 #-----
14 # Common settings
15 # https://editorconfig.org/#supported-properties
16 #-----
17
18 # This is the top most editorconfig file
19
20 root = true
21
22 # All files
23
24 [*]
25
26 charset = utf-8-bom
27 indent_style = space
28 indent_size = 4
29 insert_final_newline = false
30 trim_trailing_whitespace = true
31
32 #-----
33 # Language Conventions
34 # https://docs.microsoft.com/en-us/visualstudio/ide/editorconfig-language-conventions
35 #-----
36
37 [*.*]
38
39 # Avoid var
40 csharp_style_var_for_built_in_types = false:warning
41 csharp_style_var_when_type_is_apparent = false:warning
42 csharp_style_var_elsewhere = false:warning
43
```

Code analysis

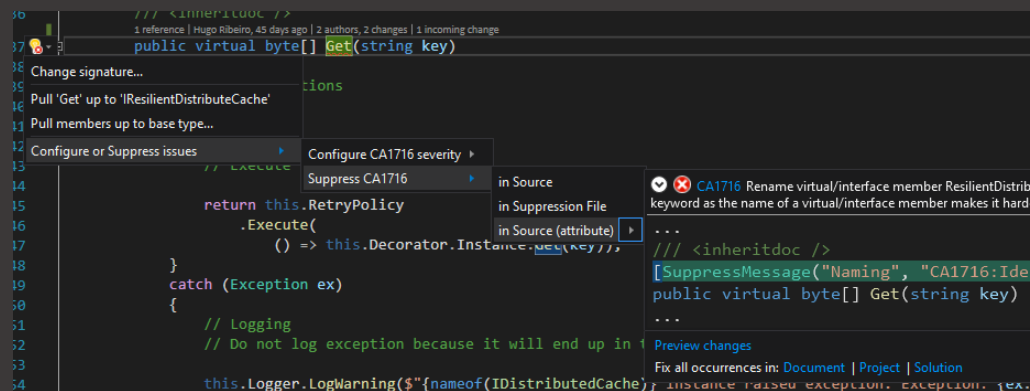
Como é evidente, a avaliação de regras de code analysis é crucial para assegurar a qualidade de qualquer componente, e ainda mais de uma biblioteca como o Hydrogen.

As regras definidas atualmente derivam diretamente das que são usadas há anos na FW Elevation e têm sido adaptadas periodicamente em função das novidades do .NET Core e das novas versões dos analyzers.

Esta é uma área em desenvolvimento ativo na Microsoft pelo que é de esperar o surgimento de cada vez mais regras, cada vez mais eficazes.

A maioria dos developers olham para este “trabalho extra” com relutância e não lhe dão a devida atenção. Mas é fundamental que se observem as melhores práticas no Hydrogen:

- A compilação da SLN não deve incluir nenhum warning (o único warning aceitável é o das assemblies externas não assinadas).
- Todos os erros e warnings de code analysis devem ser resolvidos.
- Há falsos positivos e há situações em que determinado warning é aceitável (por exemplo, ter um _ no nome de um método de teste não tem mal nenhum). Nesses casos, o erro deve ser suprimido SEMPRE usando um atributo e deve ser incluída uma justificação. Não devem ser usados #pragma nem ficheiros globais de exceção.



```
36 /// <inheritdoc />  
37 public virtual byte[] Get(string key)  
38  
39 Change signature...  
40 Pull 'Get' up to 'IResilientDistributeCache'  
41 Pull members up to base type...  
42 Configure or Suppress issues  
43 Configure CA1716 severity  
44 Suppress CA1716  
45 in Source  
46 in Suppression File  
47 in Source (attribute)  
48 return this.RetryPolicy  
49 .Execute(  
50 () => this.Decorator.Instance.Get(key))  
51 catch (Exception ex)  
52 {  
53 // Logging  
54 // Do not log exception because it will end up in 1  
55 this.Logger.LogWarning($"{nameof(IDistributedCache)}, instance raised exception, exception: {ex}");
```

```
/// <inheritdoc />  
[SuppressMessage("Microsoft.Naming", "CA1716:IdentifiersShouldNotMatchKeywords", Justification = "Imposed by IDistributedCache design.")]  
public virtual byte[] Get(string key)  
{  
    ...  
}
```