What’s New in ReSharper 9?

Dmitri Nesteruk
Developer Improvement Specialist ☺
dn@jetbrains.com @dnesteruk
skype: dmitri.nesteruk

Good Housekeeping

Unified installer

Common **Platform**

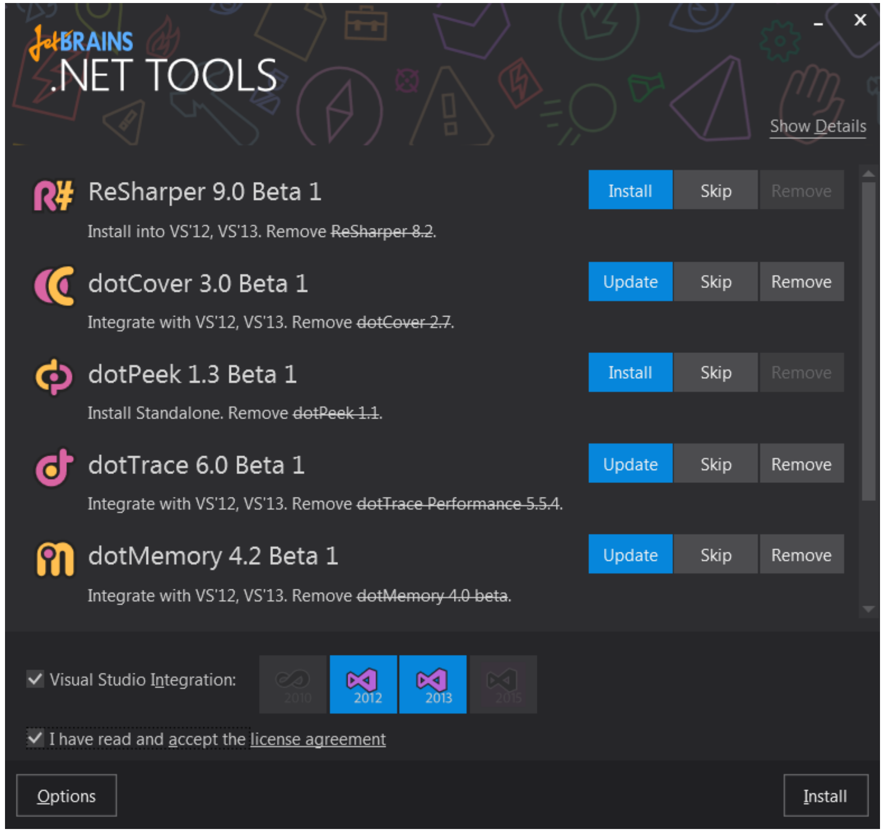
- Shared resources

Enable/disable features

Simpler licensing

- ReSharper
- ReSharper C++
- ReSharper Ultimate

soon



JetBRAINS .NET TOOLS

ReSharper 9.0 Beta 1
Install into VS'12, VS'13. Remove ~~ReSharper 8.2~~.

dotCover 3.0 Beta 1
Integrate with VS'12, VS'13. Remove ~~dotCover 2.7~~.

dotPeek 1.3 Beta 1
Install Standalone. Remove ~~dotPeek 1.1~~.

dotTrace 6.0 Beta 1
Integrate with VS'12, VS'13. Remove ~~dotTrace Performance 5.5.4~~.

dotMemory 4.2 Beta 1
Integrate with VS'12, VS'13. Remove ~~dotMemory 4.0 beta~~.

☑ Visual Studio Integration: 2010 2012 2013 2015

☑ I have read and accept the license agreement

Performance

Startup time – deferred and optimised initialisation

ReSharper Platform. Single set of resources used

“Warm storage” for caches

- Most notably SWEA
- Faster project load

Usability Improvement

Go to action (universal Alt+Enter)

Search in Options

Visual file layout editor

Regular Expressions

Syntax highlighting + brace matching

Code completion of regular expression symbols

Code completion of captured groups

Validation utility

Mark string as regex

Diagrams

Type dependency diagram

Visually navigate between dependencies of types:

- Usage, aggregation, inheritance, return type, constructor injection

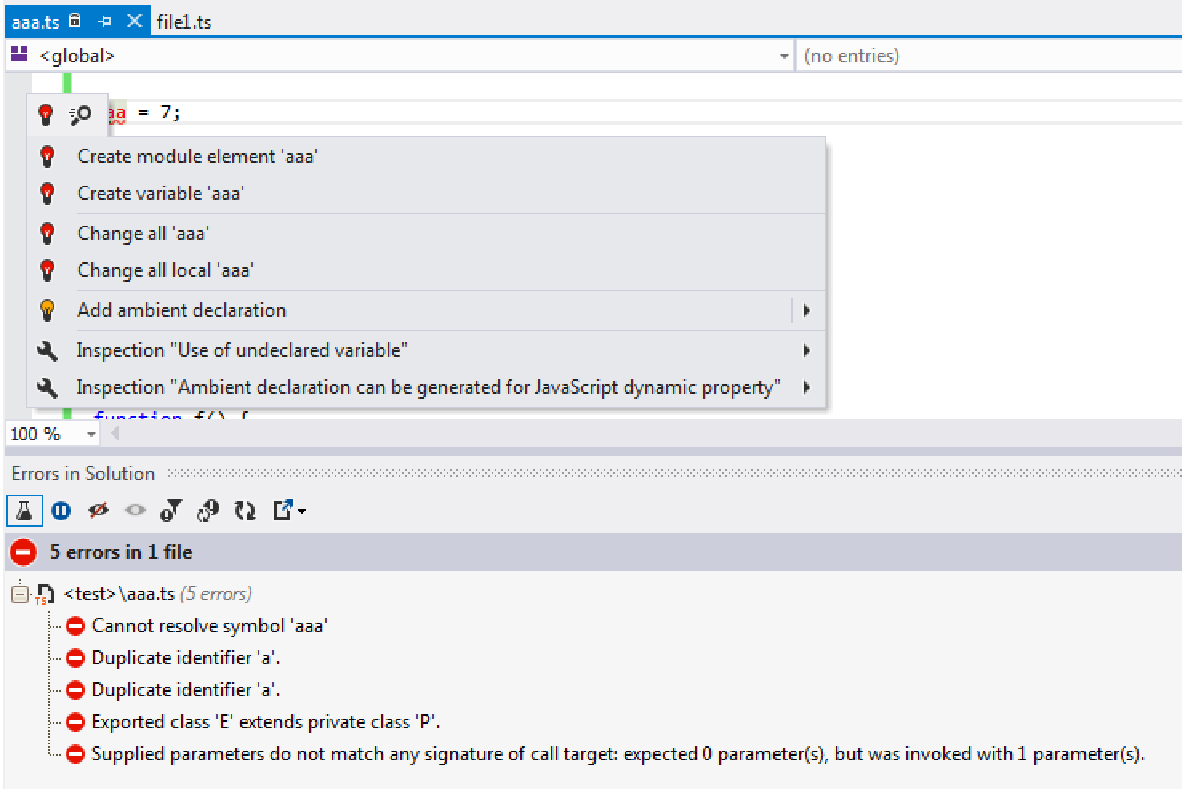
Drag and drop refactoring

- invokes move refactoring

TypeScript

270 new inspections, many with Quick Fixes

E.g. create from usage, change all, etc.



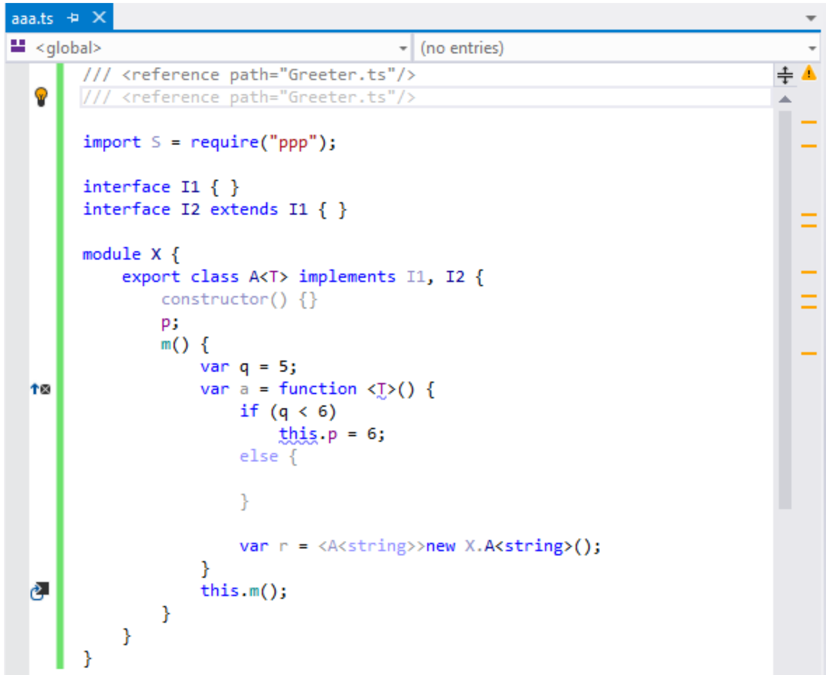
5 errors in 1 file

<test>\aaa.ts *(5 errors)*

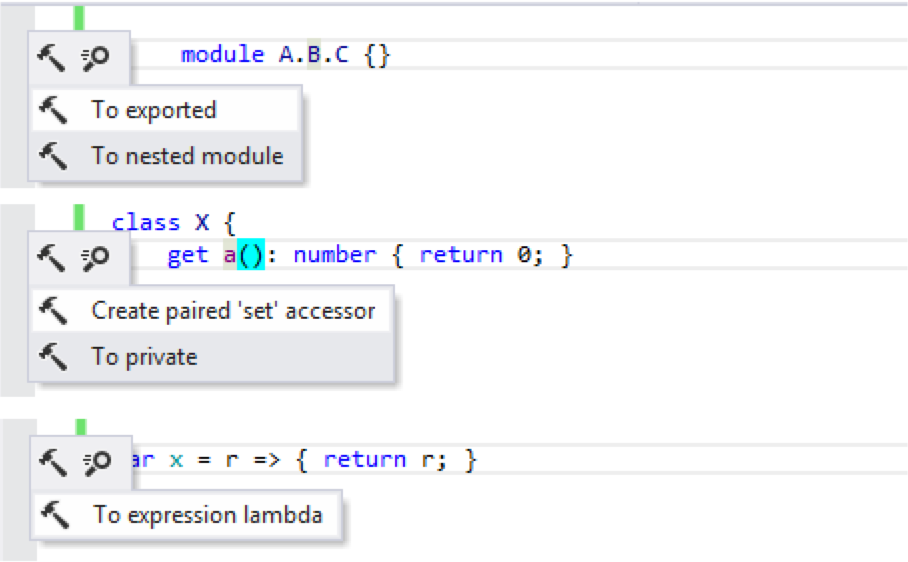
- Cannot resolve symbol 'aaa'
- Duplicate identifier 'a'.
- Duplicate identifier 'a'.
- Exported class 'E' extends private class 'P'.
- Supplied parameters do not match any signature of call target: expected 0 parameter(s), but was invoked with 1 parameter(s).

TypeScript

New inspections + context actions



```
/// <reference path="Greeter.ts"/>
/// <reference path="Greeter.ts"/>

import S = require("ppp");

interface I1 { }
interface I2 extends I1 { }

module X {
    export class A<T> implements I1, I2 {
        constructor() {}
        p;
        m() {
            var q = 5;
            var a = function <T>() {
                if (q < 6)
                    this.p = 6;
                else {

                }

                var r = <A<string>>new X.A<string>();
            }
            this.m();
        }
    }
}
```



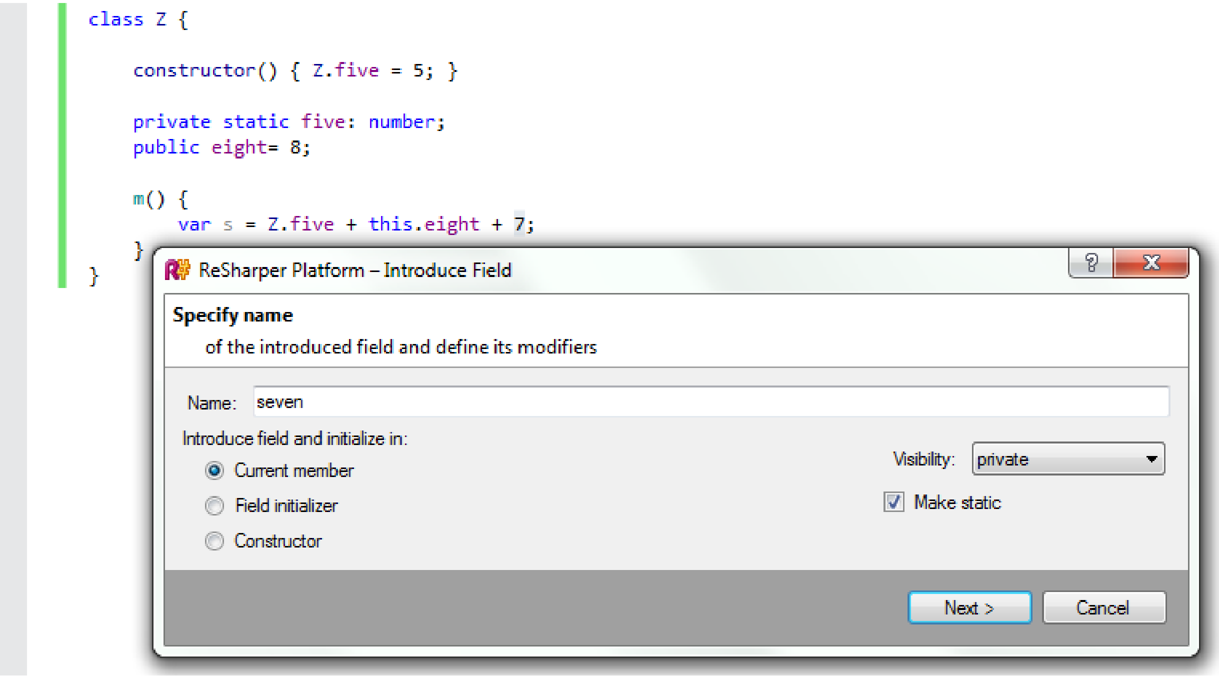
```
module A.B.C {}
```

```
class X {
    get a(): number { return 0; }
```

```
ar x = r => { return r; }
```

TypeScript

New Refactorings – Introduce Field, Copy Type, Move



```
class Z {

    constructor() { Z.five = 5; }

    private static five: number;
    public eight= 8;

    m() {
        var s = Z.five + this.eight + 7;
    }
}
```

ReSharper Platform – Introduce Field

Specify name

of the introduced field and define its modifiers

Name: seven

Introduce field and initialize in:

● Current member

○ Field initializer

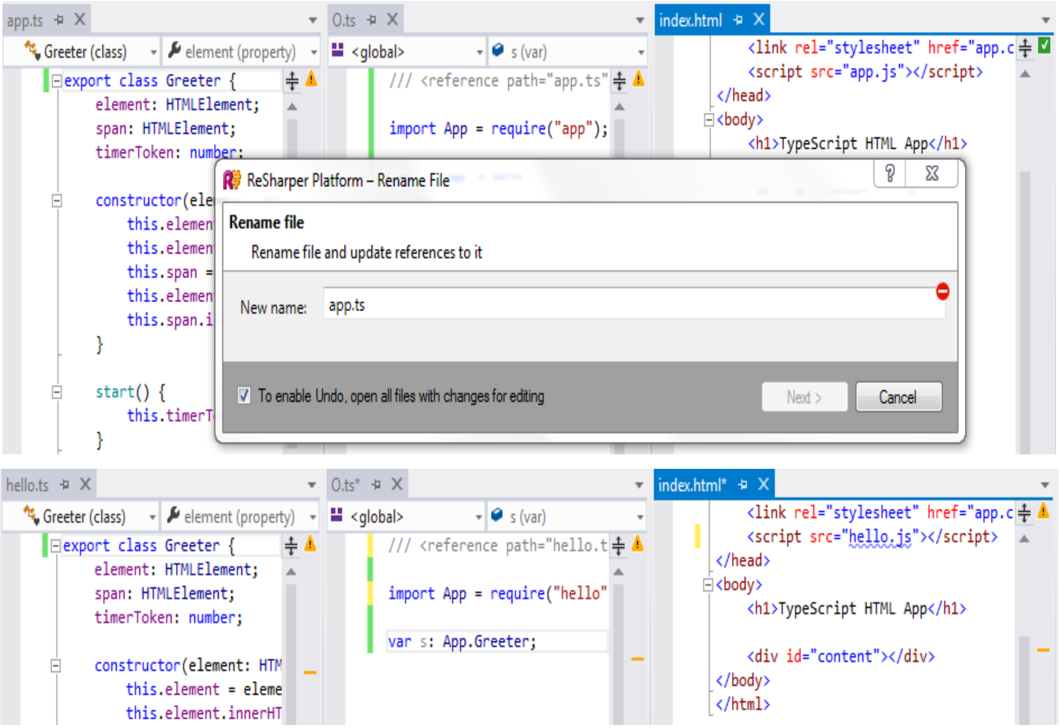
○ Constructor

Visibility: private

☑ Make static

TypeScript

Rename File – updates file based module dependencies, reference comments, etc.



```
export class Greeter {
    element: HTMLElement;
    span: HTMLElement;
    timerToken: number;

    constructor(ele
        this.elemen
        this.elemen
        this.span =
        this.elemen
        this.span.i
    }

    start() {
        this.timerT
    }
```

```
/// <reference path="app.ts"

import App = require("app");
```

```
    <link rel="stylesheet" href="app.c
    <script src="app.js"></script>
</head>
<body>
    <h1>TypeScript HTML App</h1>
```

ReSharper Platform – Rename File

Rename file

Rename file and update references to it

New name: app.ts

☑ To enable Undo, open all files with changes for editing

```
export class Greeter {
    element: HTMLElement;
    span: HTMLElement;
    timerToken: number;

    constructor(element: HTM
        this.element = eleme
        this.element.innerHT
```

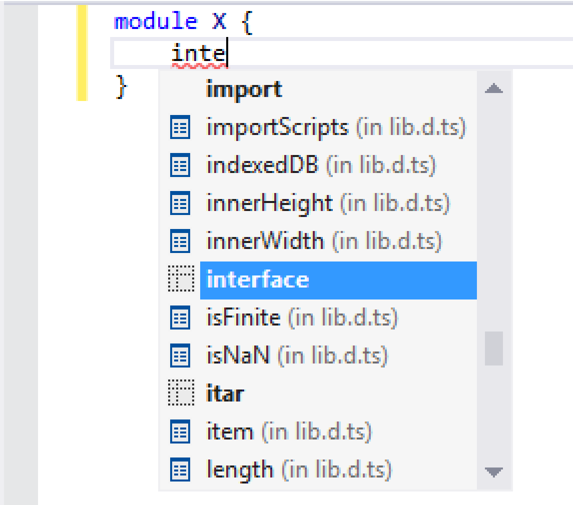
```
/// <reference path="hello.t

import App = require("hello"

var s: App.Greeter;
```

```
    <link rel="stylesheet" href="app.c
    <script src="hello.js"></script>
</head>
<body>
    <h1>TypeScript HTML App</h1>

    <div id="content"></div>
</body>
</html>
```

TypeScript

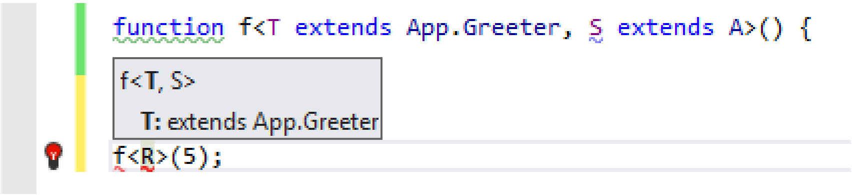
Live Templates

TS specific template scopes



```
module X {
    inte
}
```

Parameter info + generics



```
function f<T extends App.Greeter, S extends A>() {
f<R>(5);
```

Visual Studio 2015

Uses Roslyn project model for files and references

Integrates Visual Studio “bulb”

Disable Roslyn “squiggles”

ASP vNext project support

(PS. don’t forget Visual Studio 2013 Community Edition!)

C# 6.0

Done

Null propagation

Expression bodied members

Using static members

Auto property initialisers

Exception filters

Index initialisers

Future update

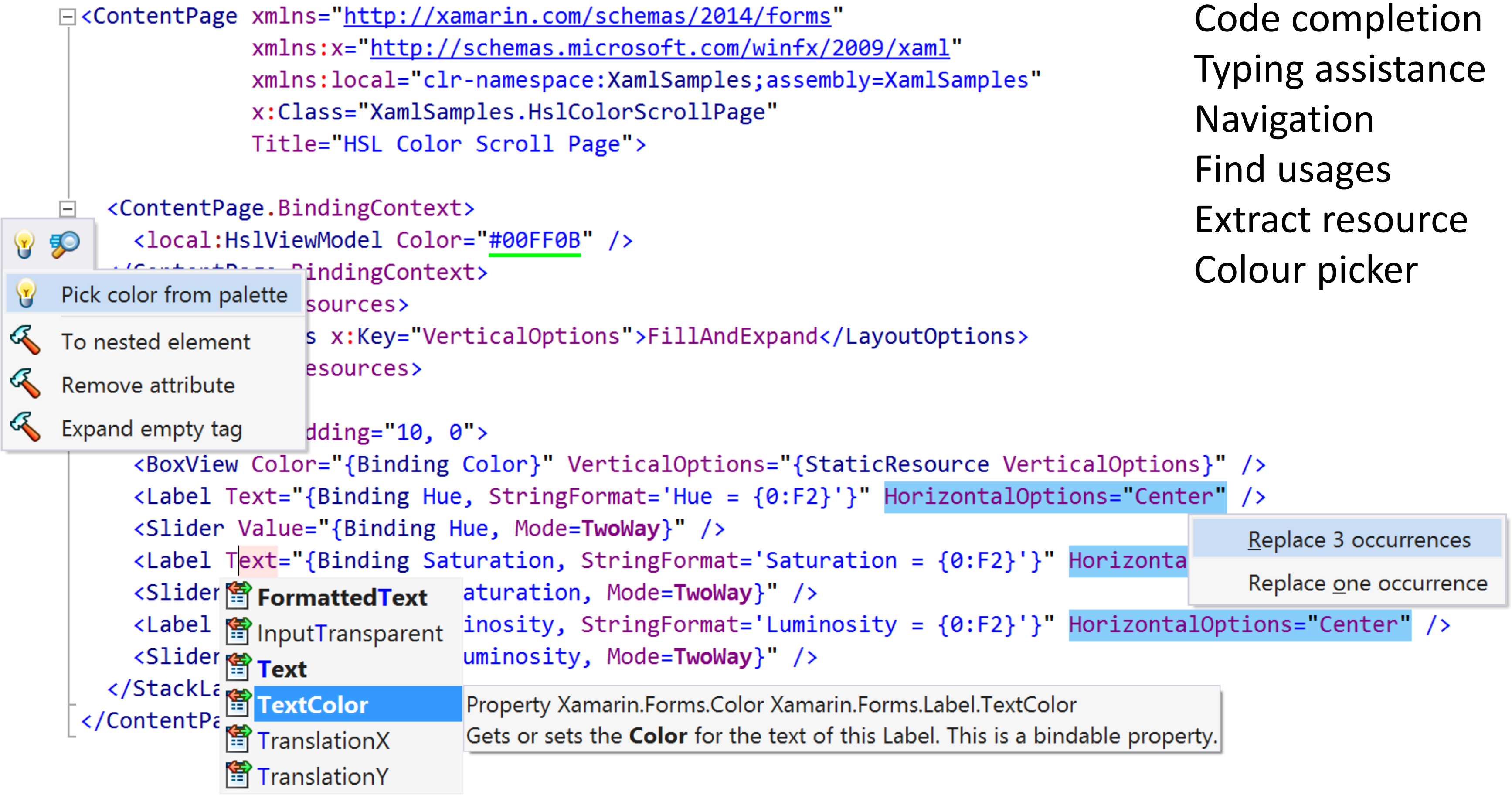
`nameof` operator

String interpolation

Further spec changes

(e.g. `using static`)

Xamarin Forms



```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:XamlSamples;assembly=XamlSamples"
             x:Class="XamlSamples.HslColorScrollPage"
             Title="HSL Color Scroll Page">

  <ContentPage.BindingContext>
    <local:HslViewModel Color="#00FF0B" />
  </ContentPage.BindingContext>
                 sources>
                 s x:Key="VerticalOptions">FillAndExpand</LayoutOptions>
                 esources>

                 dding="10, 0">
    <BoxView Color="{Binding Color}" VerticalOptions="{StaticResource VerticalOptions}" />
    <Label Text="{Binding Hue, StringFormat='Hue = {0:F2}'}" HorizontalOptions="Center" />
    <Slider Value="{Binding Hue, Mode=TwoWay}" />
    <Label Text="{Binding Saturation, StringFormat='Saturation = {0:F2}'}" Horizonta
    <Slider      aturation, Mode=TwoWay}" />
    <Label       inosity, StringFormat='Luminosity = {0:F2}'}" HorizontalOptions="Center" />
    <Slider      uminosity, Mode=TwoWay}" />
  </StackLa
</ContentPa
```

Property Xamarin.Forms.Color Xamarin.Forms.Label.TextColor
Gets or sets the **Color** for the text of this Label. This is a bindable property.

- Code completion
- Typing assistance
- Navigation
- Find usages
- Extract resource
- Colour picker

What’s new?

ReSharper Platform

Navigation

“ReSharper for functionality”

- Go to Action, Search in options, easy code formatter configuration

Editing

Inspections

Code Style

Visual file layout editor

Type Dependency Diagram

Xamarin Forms

Shared Projects improvements

Visual Studio 2015

Languages

- C# 6.0
- C++
- TypeScript
- Regular Expressions

https://www.jetbrains.com/resharper

Dmitri Nesteruk
dn@jetbrains.com
@dnesteruk
upsource.jetbrains.com