

WHAT'S NEW IN C# 15

Damir Arh, Razum d.o.o.

Microsoft MVP

ABOUT ME

- Razum d.o.o.
- Microsoft MVP
- <https://damirscorner.com>
- @DamirArh@mas.to
- [@damirscorner.com](https://damirscorner.com)
- [@DamirArh](https://DamirArh)



A QUICK OVERVIEW

- exhaustiveness checking
 - closed class hierarchies
 - union types
- syntax simplifications
 - collection expression arguments
 - extension indexers
 - labeled break and continue
- groundwork for future versions
 - unsafe evolution
 - runtime async

CURRENT AVAILABILITY

- IDE support
 - Visual Studio 2026 Insiders
 - JetBrains Rider 2026.3 EAP
 - Visual Studio Code
 - C# Dev Kit Prerelease
 - ReSharper
- Prerequisites
 - .NET 11 SDK RC1

DISCRIMINATING BETWEEN TYPES

- Common handling of multiple types before C# 15
 - Derive from (abstract) base class
 - Implement (marker) interface
 - Unrelated types
- **Pattern matching** to branch code based on type

NON-EXHAUSTIVE PATTERN MATCHING

Demo

COMPILE-TIME EXHAUSTIVENESS CHECK

- No warning:

```
return shape switch
{
    Square square => Math.Pow(square.Length, 2),
    Circle circle => Math.PI * Math.Pow(circle.Radius, 2),
};
```

- Introduces warning:

```
public record Rectangle(double Length, double Width) : Shape;
```

CLOSED CLASS HIERARCHIES

```
public closed record Shape { }
```

- Exhaustiveness checking for related classes
- Base class `closed` instead of `abstract`
- Can only be derived from in the same assembly
- No warning without discard case
- Deriving another type generates a warning

CLOSED CLASS HIERARCHIES

Demo

UNION TYPES

```
public union Shape(Square, Circle);
```

- Exhaustiveness checking for unrelated classes
- New `union` keyword
- Explicitly defines allowed types
- No warning without discard case
- Adding another type generates a warning

UNION TYPES

Demo

UNION TYPE INTERNALS

- Marked with `[Union]` attribute
- Implements `IUnion` interface
 - Single member: `Value` property
- For each case type
 - Implements a constructor
 - Implements an implicit conversion operator
- Can contain custom members
- Allows a custom implementation

CUSTOM UNION TYPES

Demo

COLLECTION EXPRESSION ARGUMENTS

- C# 12: alternative syntax for collection initialization

```
var collectionInitializer = new[] { 1, 2, 3 };  
int[] collectionExpression = [1, 2, 3];
```

- C# 15: support for collection expression arguments

```
List<int> withConstructorArgument = [with(10), 1, 2, 3];
```

COLLECTION EXPRESSION ARGUMENTS

Demo

EXTENSION INDEXERS

- C# 3
 - extension methods
- C# 14 (new syntax)
 - instance and static extension methods
 - instance and static extension properties
 - extension operators
- C# 15
 - extension indexers

EXTENSION INDEXERS

Demo

LABELED BREAK AND CONTINUE

- C# 1

```
goto label  
// ...  
label:
```

- C# 15

```
label: foreach (var item in items)  
{  
    // ...  
    continue label;  
    // ...  
    break label;  
}
```

LABELED BREAK AND CONTINUE

Demo

ORIGINAL UNSAFE MODEL

- Introduced in C# 1
- Unsafe context allows unverifiable code
- Any pointer features are allowed only inside one
- Compiler option required to compile unsafe code

UPDATED UNSAFE MODEL

- Introduced in C# 15 as a preview feature
- Declaring pointers isn't considered unsafe anymore
- Unsafe context now propagates to caller
- Member will be able to validate safety instead
- Project will have to opt-in to new model

```
<PropertyGroup>  
  <LangVersion>preview</LangVersion>  
</PropertyGroup>
```

COMPILER-GENERATED ASYNC/AWAIT

- Introduced in C# 5
- Compiler generates state-machine classes
- Runtime has no awareness of async methods
- State-machine infrastructure visible in call stacks

RUNTIME-NATIVE ASYNC/AWAIT

- Experimentally introduced in .NET 10
- Requires opt-in in .NET 11
- All .NET runtime libraries already opted-in
- Runtime tracks asynchronous execution
- Compiler only adds attributes and async helper calls
- No additional artifacts in call stacks

```
<PropertyGroup>  
  <Features>runtime-async=on</Features>  
</PropertyGroup>
```

LINKS

- damirscorner.com/link/
 - [Cs15GitHub](#)
 - [Cs15Blogposts](#)
 - [Cs15Docs](#)
 - [Cs15Status](#)