What's New in C#?

IMPROVING YOUR CODE AND MAKING YOUR JOB EASIER

Brendan Enrick

Twitch.tv/DevChatter

@Brendoneus

Agenda

Current C# Highlights

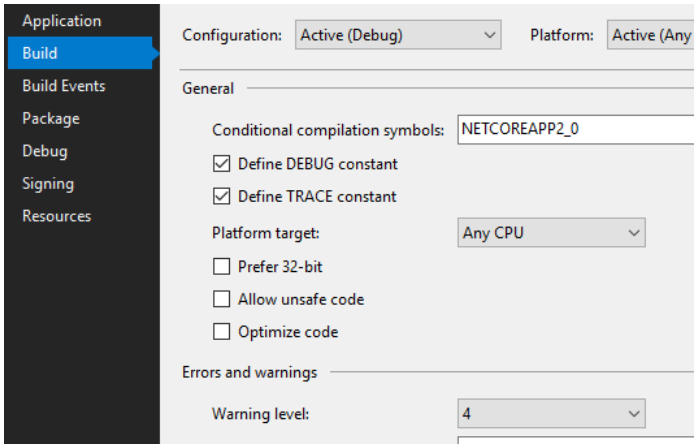
Preview of Features Coming in C# 8

Dev Chatter

Selecting C# Version

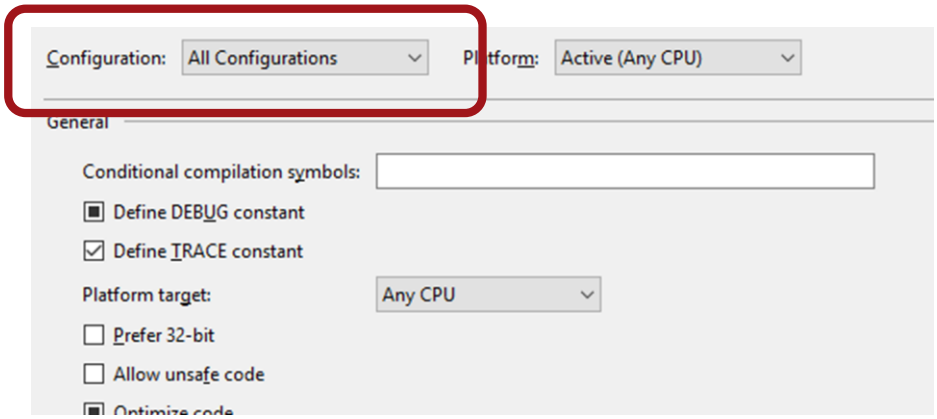
Dev Chatter

Navigate to Properties Build Tab



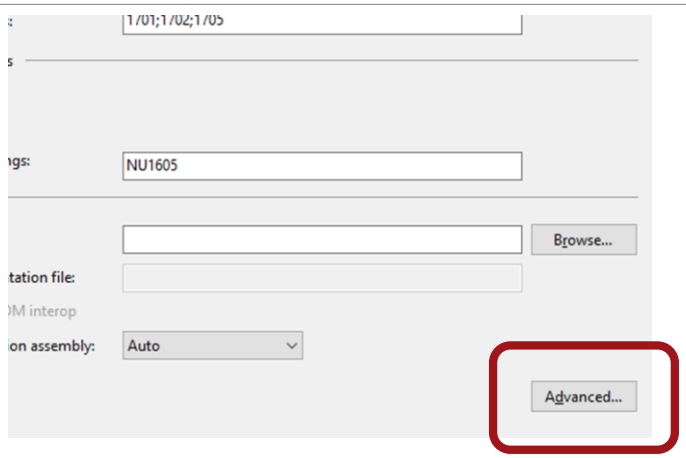
Dev Chatter

Change Configuration to All



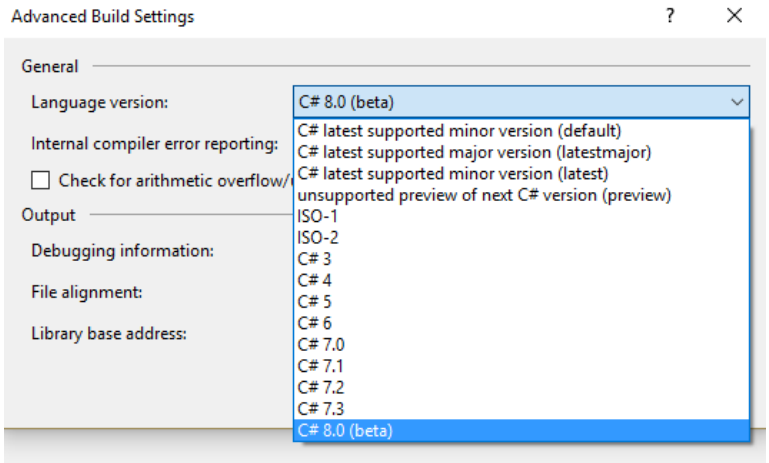
Dev Chatter

Open the Advanced Menu



Dev Chatter

Select Version



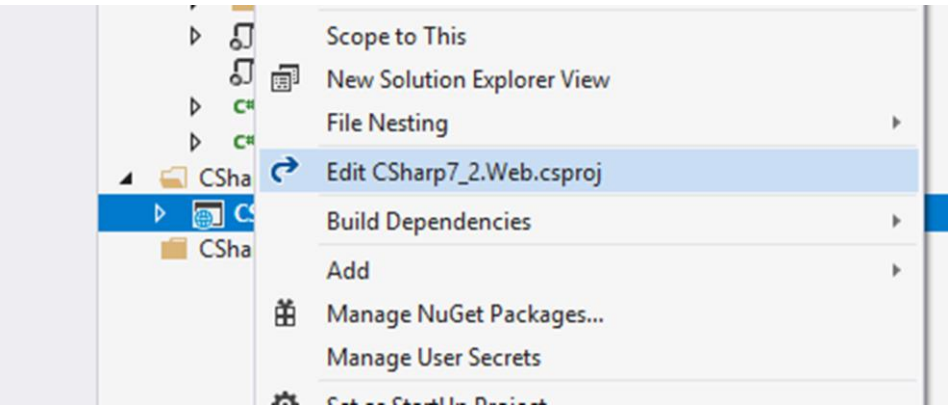
Dev Chatter

Version Choices

C# latest supported minor version (default)
C# latest supported major version (latestmajor)
C# latest supported minor version (latest)
C# 7.0
C# 7.1
C# 7.2
C# 7.3
C# 8.0 (beta)

Dev Chatter

Edit Project File



Dev Chatter

Edit Project File

```
<PropertyGroup>
 <LangVersion>8.0</LangVersion>
</PropertyGroup>
```

Dev Chatter

Note for ASP.NET Core 2.0 Users

Razor pages don't correctly read the C# language version when compiling.

- C# 7.1, 7.2, and C# 7.3 code cannot be in the razor page itself.

Dev Chatter

ASP.NET Core 2.0 Workaround

```
public class FixedCSharpCompiler : CSharpCompiler
{
    public FixedCSharpCompiler(RazorReferenceManager manager,
        IHostingEnvironment hostingEnvironment)
        : base(manager, hostingEnvironment)
    {
    }

    public override CSharpParseOptions ParseOptions =>
        base.ParseOptions.WithLanguageVersion(LanguageVersion.Latest);
}
```

Dev Chatter

Current C# Highlights

IMPROVE YOUR CODE TODAY WITH THESE

Dev Chatter

Auto Property Initializers

```
// Previous C# 6 Auto-Property Initializer
public string FirstName { get; set; } = "Viq";
```

Dev Chatter

Readonly Properties

```
// With C# 6 Readonly Auto-Property
public string FirstName { get; } = "Viq";
```

Dev Chatter

Readonly Properties

```
public AutoPropPerson()
{
    FirstName = "Viq";
}

public string FirstName { get; }
```

Dev Chatter

Expression-Bodied Methods

```
public class Rectangle
{
    public int Length { get; set; }
    public int Width { get; set; }
    public bool IsSquare()
    {
        return Length == Width;
    }
}
```

Dev Chatter

Expression-Bodied Methods

```
public class Rectangle
{
    public int Length { get; set; }
    public int Width { get; set; }
    public bool IsSquare() => Length == Width;
}
```

Dev Chatter

Expression-Bodied Properties

```
public class Rectangle
{
    public int Length { get; set; }
    public int Width { get; set; }
    public int Area
    {
        get { return Length * Width; }
    }
}
```

Dev Chatter

Expression-Bodied Properties

```
public class Rectangle
{
    public int Length { get; set; }
    public int Width { get; set; }
    public int Area => Length * Width;
}
```

Dev Chatter

Other Expression-Bodied Members

Constructors

Finalizers

Getters

Setters

Switch Cases (in C# 8)

Dev Chatter

String Interpolation

```
public class Person
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string FullName
    {
        get { return $"{FirstName} {LastName}"; }
    }
}
```

Dev Chatter

String Interpolation

```
public class Person
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string FullName => $"{FirstName} {LastName}";
}
```

Dev Chatter

nameof Expressions

```
throw new ArgumentException(
                $"{nameof(id)} must be set", nameof(id));
```

Dev Chatter

Null Conditionals (Null Propagation)

```
public string GetNicknameByUserId(Guid userId)
{
    var user = _dataStore.UserById(userId);
    if (user == null)
    {
        return string.Empty;
    }
    return user.Nickname;
}
```

Dev Chatter

Null Conditionals (Null Propagation)

```
public string GetNicknameByUserId(Guid userId)
{
    var user = _dataStore.UserById(userId);
    return user?.Nickname;
}
```

Dev Chatter

Null Conditionals (Null Propagation)

```
public string GetNicknameByUserId(Guid userId)
{
    var user = _dataStore.UserById(userId);
    return user?.Nickname ?? "";
}
```

Dev Chatter

Null Conditionals (Null Propagation)

```
public string GetFirstChildName()
{
    return this.Children?[0]?.Name;
}
```

Dev Chatter

Output Parameters (Not Variables)

```
// C# 6 and earlier
int inputInt;
if (int.TryParse(rawInput, out inputInt))



// C# 7
if (int.TryParse(rawInput, out int inputInt))
```

Dev Chatter

Discards

```
public bool IsValidEnum(string text)
{
    return Enum.TryParse(text, out DemoEnum _);
}
```

Dev Chatter

Pattern Matching

Dev Chatter

Pattern Matching – Is Expressions

```
public static int CalculateArea(Shape shape)
{
    if (shape is null) return 0;
    if (shape is Rectangle rec)
        return rec.Length * rec.Width;
    if (shape is Triangle tri)
        return tri.Base * tri.Height / 2;
    return 0;
}
```

Pattern Matching – Is Expression

```
Rectangle rec = shape as Rectangle;
if (rec != null)
    return rec.Length * rec.Width;
```

Pattern Matching – Is Expression

```
if (shape is Rectangle rec)
    return rec.Length * rec.Width;
```

Dev Chatter

Pattern Matching – Switch Statements

```
switch (shape) {
    case Rectangle rec:
        WriteLine($"Rectangle: {rec.Length} x {rec.Width}");
        break;
    case Triangle tri:
        WriteLine($"Triangle: b-{tri.Base} h-{tri.Height}");
        break;
    case null:
        WriteLine("null");
        break;
    default:
        WriteLine("default");
        break;
}
```

Dev Chatter

Pattern Matching – Switch Statements

```
switch (shape)
{
    case Rectangle sq when sq.Length == sq.Width && sq.Length > 150
            && !knownSquares.All(s => s.Length > 7 && s.Length % 2 != 0):
        WriteLine($"Square: {sq.Length}");
        break;
    case Rectangle bigRec when bigRec.Length > 100:
        WriteLine($"Big Rectangle: L-{bigRec.Length}");
        break;
    case Rectangle rec:
        WriteLine($"Rectangle: {rec.Length} x {rec.Width}");
        break;
}
```

Dev Chatter

Pattern Matching – Switch Statements

```
switch (shape)
{
    case Rectangle sq when SquareMatchesMyCriteria(sq):
        WriteLine($"Square: {sq.Length}");
        break;
    case Rectangle bigRec when bigRec.Length > 100:
        WriteLine($"Big Rectangle: L-{bigRec.Length}");
        break;
    case Rectangle rec:
        WriteLine($"Rectangle: {rec.Length} x {rec.Width}");
        break;
}
```

Dev Chatter

Pattern Matching – Action Filter

```
[AccountAuthorize("accountId")]

[AccountAuthorize("account")]

[AccountAuthorize("accountIds")]

[AccountAuthorize("accounts")]
```

Dev Chatter

Pattern Matching – Action Filter

```
public override void OnActionExecuting(HttpActionContext actionContext) {
    switch (actionContext.ActionArguments[ArgName]) {
        case int accountId:
            AuthorizeById(accountId);
            break;
        case IEnumerable<AccountEntity> accountEnumerable:
            AuthorizeByEnumerable(accountEnumerable);
            break;
        case AccountEntity account:
            AuthorizeByAccount(account);
            break;
        default:
            throw new AuthorizationException(ArgName);
    }
}
```

Dev Chatter

Safe Null Check

```
if (myVariable is null)
    // Do Stuff Here
```

Dev Chatter

Tuples

Dev Chatter

Tuple Nuget Package

System.ValueTuple

```
> Install-Package System.ValueTuple
```

Not required after .NET 4.7

Dev Chatter

Tuple Example

```
var result = SafeParse("123");
Console.WriteLine(result.number);
Console.WriteLine(result.success);

(bool success, int number) SafeParse(string s)
    => (int.TryParse(s, out int n), n);
```

Dev Chatter

Tuple Deconstruction Example

```
var (number, success) = SafeParse("123");
Console.WriteLine(number);
Console.WriteLine(success);

(bool success, int number) SafeParse(string s)
    => (int.TryParse(s, out int n), n);
```

Dev Chatter

Tuple One-Line Assignment

```
private int _x;
private int _y;
private int _z;
public Coord3D(int x, int y, int z)
{
    (_x, _y, _z) = (x, y, z);
}
```

Dev Chatter

Tuple Related Data

```
public class SpriteData
{
    public SpriteData(int x, int y, string name)
    {
        (X, Y) = (x, y);
        Name = name;
    }

    public int X { get; }
    public int Y { get; }
    public string Name { get; }
}
```

Dev Chatter

Tuple with Discards

```
public string ShowCoordinates(Guid id)
{
    (int x, int y, _) = GetPosition(id);
    return $"({x},{y})";
}

public (int, int, string) GetPosition(Guid id)
{
    // Pretend this gets the item
    return (1, 2, "foo");
}
```

Dev Chatter

C# 7.1 - 7.3 Highlights

Dev Chatter

async Main (C# 7.1)

```
public class Program
{
    public static async Task<int> Main(string[] args)
    {
        BuildWebHost(args).Run();
        return await DoNothing();
    }

    private static async Task<int> DoNothing()
    {
        await Task.Delay(1000);
        return 0;
    }
}
```

Dev Chatter

Inferred Tuple Names (C# 7.1)

```
int count = 5;
string label = "Colors used in the map";
var pair = (count, label);
return $"pair.count:{pair.count} and
pair.label:{pair.label}";
```

Dev Chatter

default Literal Expressions (C# 7.1)

```
string oldWay = default(string);

string newWay = default;
```

Dev Chatter

Non-Trailing Named Arguments (C# 7.2)

```
public string GetFullName()
{
    return Name("Brendan", middleName: "Danger", "Enrick");
}
```

Dev Chatter

Num Literals - Leading Underscores (C# 7.2)

```
public string GetNewBinaryLiteral()
{
    int newWay = 0b_0111_1110;

    return $"newWay(0b_0111_1110):{newWay}";
}

public string GetNewHexLiteral()
{
    int newWay = 0x_F0_F0_F0;

    return $"newWay(0x_F0_F0_F0):{newWay}";
}
```

Dev Chatter

private protected Access Modifier (C# 7.2)

```
public class ParentClass
{
    private protected string text = "parent";
}

public class ChildClass : ParentClass
{
    public string GetThatValue()
    {
        base.text = "from child";
        return text.ToString();
    }
}
```

Dev Chatter

Tuple Operator Equality (C# 7.3)

```
("Stir", "Trek") == ("Stir", "Trek"); // True
("Stir", "Trek") == ("Brendan", "Enrick"); // False


("brendan", "Enrick") != ("Brendan", "Enrick"); // True
("Brendan", "Enrick") != ("Brendan", "Enrick"); // False
```

Dev Chatter

Enum Type Constraint (C# 7.3)

```
private string GetName<T>(T obj) where T : System.Enum
{
    return System.Enum.GetName(obj.GetType(), obj);
}
```

Dev Chatter

Looking Toward C# 8

THIS SECTION ISN'T FINAL. ANY OF IT COULD CHANGE!

Dev Chatter

Reverse Indexing (Hat Operator)

A	B	C	D	E	F	G	H	I
0	1	2	3	4	5	6	7	8
^9	^8	^7	^6	^5	^4	^3	^2	^1

Dev Chatter

Range Indexing

```
var numbers = new[] {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

var middle = numbers[3..7];
var firstFew = numbers[..3];
var lastBunch = numbers[4..];

var lastFew = numbers[^3..];
var lastElement = numbers[^1];
```

Dev Chatter

Nullable Context - Project

```
<NullableContextOptions>enabled</NullableContextOptions>
```

Dev Chatter

Nullable Context - Processor Directives

```
#nullable enable

public class Person
{
}

#nullable disable
```

Dev Chatter

Nullable Reference Types

ALLOW NULL

Can set `null` value.

Can use default, `null` for reference types.

Access variable only where compiler can be sure it's safe.

DISALLOW NULL

Requires variable set to non-null value.

All assignments to `null` blocked.

Dev Chatter

Nullable Reference Type Syntax

```
public class Person
{
    public string FirstName;
    public string? MiddleName;
    public string LastName;

    Person() // Warning: Must set FirstName, LastName
    {
    }
}
```

Dev Chatter

Nullable Reference Type Syntax

```
public class Person
{
    public string FirstName;
    public string? MiddleName;
    public string LastName;

    public Person(string fistName, string lastName)
    {
        FirstName = firstName;
        LastName = lastName;
    }
}
```

Dev Chatter

Null Forgiving Operator

```
public int GetMiddleNameLength()
{
    // Warning of possible null.
    return MiddleName.Length;
}
```

Dev Chatter

Null Forgiving Operator

```
public int GetMiddleNameLength()
{
    // Bad Code: Don't do this!
    return MiddleName!.Length;
}
```

Dev Chatter

Pattern Matching Improvements

Switch Expressions

Property Patterns

Tuple Patterns

Dev Chatter

Switch Expressions (Before)

```
public Villains GetVillainByMovie(Movies movie)
{
    switch (movie)
    {
        case Movies.Avengers:
            return Villains.Loki;
        case Movies.AgeOfUltron:
            return Villains.Ultron;
        case Movies.InfinityWar:
            return Villains.Thanos;
        case Movies.EndGame:
            return Villains.ProbablyThanos;
        default:
            throw new ArgumentException("Invalid Movie", nameof(movie));
    }
}
```

Dev Chatter

Switch Expressions

```
public Villains GetVillainByMovie(Movies movie)
{
    return movie switch
    {
        Movies.Avengers => Villains.Loki,
        Movies.AgeOfUltron => Villains.Ultron,
        Movies.InfinityWar => Villains.Thanos,
        Movies.EndGame => Villains.ProbablyThanos,
        _ => throw new ArgumentException("Invalid Movie", "movie"),
    };
}
```

Dev Chatter

Property Patterns

```
public Receipt HandleConferenceSale(Conference conference)
{
    return conference switch
    {
        { State: "OH", Name: "StirTrek" } => BuyNow(conference),
        { State: "OH" } => BuyTicket(conference),
        { State: "FL" } => RandomChoice(conference),
        { State: "WA" } => RandomChoice(conference),
        { State: "CA" } => RandomChoice(conference),
        _ => Receipt.None
    };
}
```

Dev Chatter

Tuple Patterns

```
public string GetResult(string choice1, string choice2)
{
    return (choice1, choice2) switch
    {
        ("rock", "scissors") => "Rock crushes scissors.",
        ("paper", "rock") => "Paper covers rock.",
        ("scissors", "paper") => "Scissors cut paper.",
        ("rock", "paper") => "Paper covers rock.",
        ("paper", "scissors") => "Scissors cut paper.",
        ("scissors", "rock") => "Rock crushes scissors.",
        (_, _) => "Tie game." // or use _ instead of (_, _) if desired.
    };
}
```

Dev Chatter

Using Declaration Scope

```
public Student GetTopStudent()
{
    const string sql = "SELECT TOP 1 * FROM [Students]";
    using var db = new SqlConnection(connectionString);
    using var cmd = new SqlCommand(sql, db);
    using SqlDataReader reader = cmd.ExecuteReader();

    // Do stuff

    return studentFromDb;
}
```

Dev Chatter

Static Local Functions

```
public double GetCurrentArea()
{
    double area = CalcArea(myRadius);
    return area;

    static double CalcArea(int r) => Math.PI * r * r;
}
```

Dev Chatter

Default Interface Methods

```
interface IBotCommand
{
    IList<string> Words { get; }
    Task Execute();
}

public class HypeCommand : IBotCommand
{
    public IList<string> Words { get; } = new[] {"Hype"};

    public Task Execute()
    {
        // Do Stuff
    }
}
```

Dev Chatter

Default Interface Methods

```
interface IBotCommand
{
    IList<string> Words { get; }
    bool ShouldExecute(string word) => Words.Any(w => w == word);
    Task Execute();
}

public class HypeCommand : IBotCommand
{
    public IList<string> Words { get; } = new[] {"Hype"};

    public Task Execute()
    {
        // Do Stuff
    }
}
```

Dev Chatter

Questions?

Dev Chatter

Thanks for spending your morning here!

Brendan Enrick

Twitch.tv/DevChatter

@Brendoneus

Dev Chatter