October 23, 2019

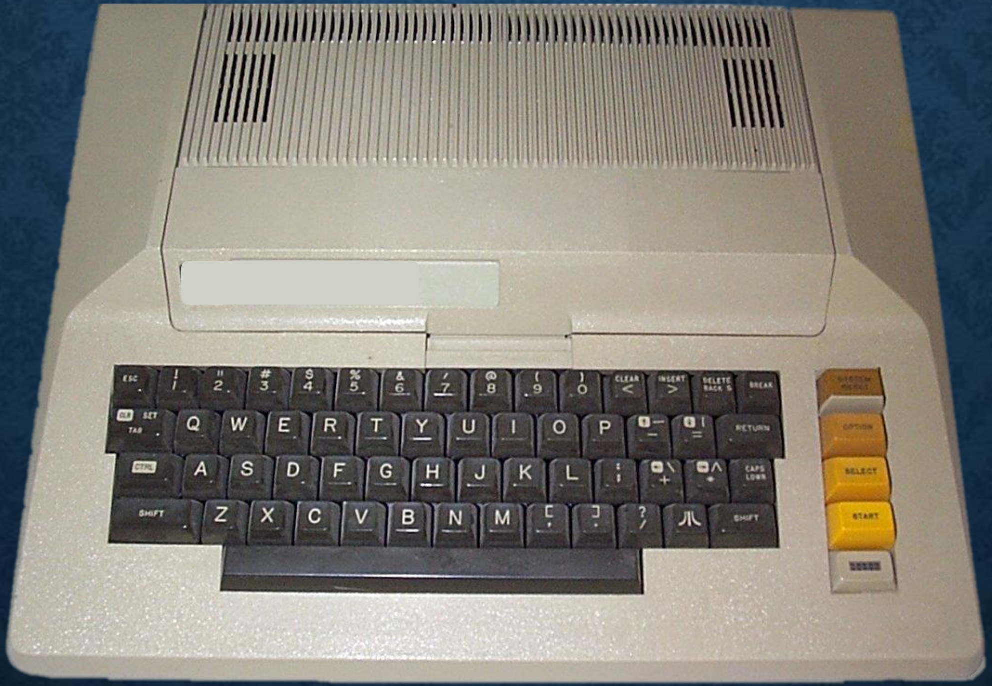
.NET Core 3

with Grant Erickson

Microsoft Partner
Microsoft

Gold DevOps
Gold Cloud Platform
Silver Application Development

WHAT IS THIS?



How much can you know?

8 Bit Legacy: The Curious History of Video Games

IntelliTect

HIGHLIGHTS

- Evolutionary Change
- C# 8 goodness
- Native JSON
- EF Enhancements
- ASP.NET Core improvements
- Stand-alone executables

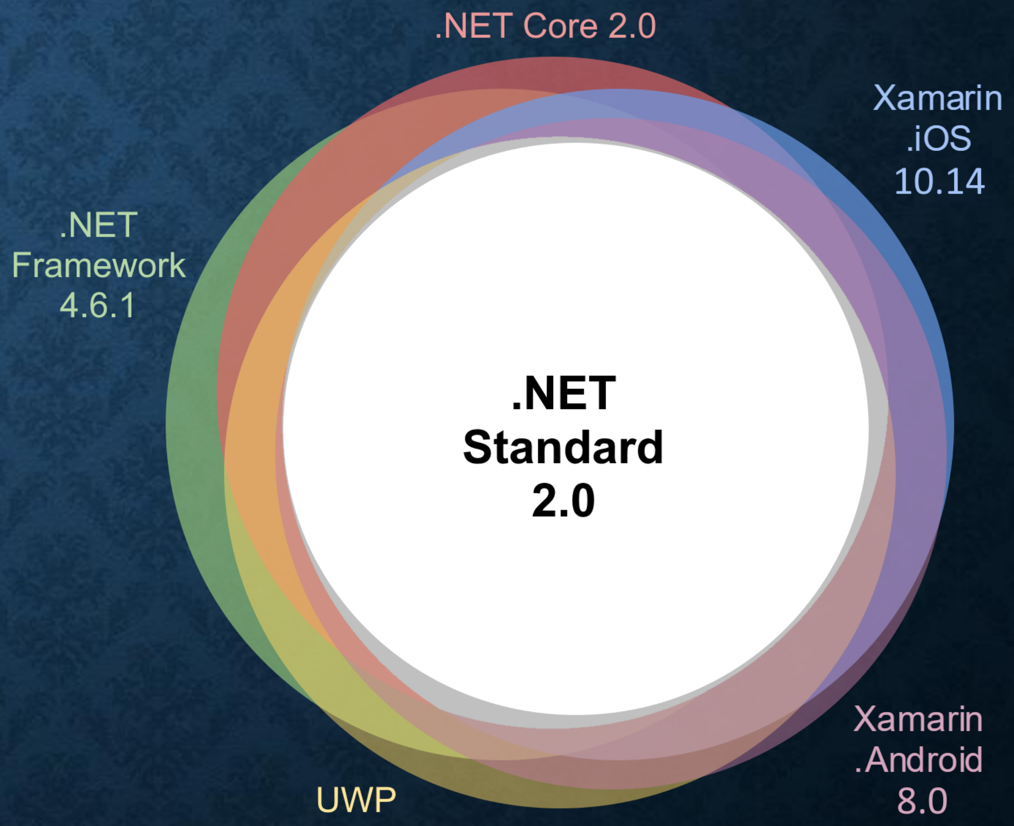
VERSIONING

Change	.NET Core Runtime	.NET Core SDK (*)
Initial release	2.2.0	2.2.100
SDK Patch	2.2.0	2.2.101
Runtime and SDK Patch	2.2.1	2.2.102
SDK Feature change	2.2.1	2.2.200

- **Major**: Significant changes occur to the product, or a new product direction. Breaking changes were taken.
- **Minor**: Public API surface area is added. A new behavior is added.
- **Patch**: Bug fixes are made. Support for a newer platform is added.

.NET STANDARD

.NET Core	.NET Standard
1.0	up to 1.6
2.0	up to 2.0
2.1	up to 2.0
2.2	up to 2.0
3.0	up to 2.1



IntelliTect

VERSIONS AND LINKING

- Version Call Differences
- Single File Executables

```
<PropertyGroup>
  <RuntimeIdentifier>win10-x64</RuntimeIdentifier>
  <PublishSingleFile>true</PublishSingleFile>
</PropertyGroup>
-- OR --
dotnet publish -r win10-x64 -p:PublishSingleFile=true
```

TREE SHAKING

- XML

```
<PropertyGroup>
  <PublishTrimmed>true</PublishTrimmed>
</PropertyGroup>
```

- CLI

```
dotnet publish -r <rid> -c Release
```

COMPILATION, BUILDS AND .NET CORE VERSION

- Tiered
    - Default is on
    - Options
        - lower-quality-but-faster
        - higher-quality-but-slower
    - Ready to Run Images
        - Natively compiled
        - Limited cross-targeting
- Build copies NuGet dependencies
- Major-version Roll Forward

IntelliTect

INDEXES AND RANGES

- Index i1 = 3 // 4th element from beginning
- Index i5 = ^1 // Last element in collection
- Index i9 = ^2 // 2nd element from the end

- Range r1 = 0..3 // First three elements
- Range r2 = 1..^0 // All elements except the first
- Range r3 = 0..^2 // All elements except the last two

ASYNC STREAMS

- Ability to await foreach

```
async IAsyncEnumerable<int> GetBigResultsAsync()
{
    await foreach (var result in GetResultsAsync())
    {
        if (result > 20) yield return result;
    }
}
```

IEEE FLOATING-POINT IMPROVEMENTS

- BitIncrement(Double) and BitDecrement(Double)
- MaxMagnitude(Double, Double) and MinMagnitude(Double, Double)
- Plus several more esoteric ones that are more accurate and IEEE compliant

JSON GOODNESS

- Utf8JsonReader & Utf8JsonWriter
  - Used for building JSON Parsers
  - 2x faster than Json.Net
  - UTF8 rather than UFT16
- JsonDocument: Reader
- JsonSerializer with deserializer
- Newtonsoft.Json has been removed
- Serialization challenges with nullable
- Reference loops are not baked

HTTP/2 SUPPORT

- Data compression of HTTP headers
- HTTP/2 Server Push
- Fixing the head-of-line blocking problem in HTTP 1.x
- Multiplexing multiple requests over a single TCP connection
- Enabled by Default (negotiation)

- TLS 1.3 & OpenSSL 1.1.1 on Linux

CRYPTOGRAPHY

- Cryptographic ciphers
    - AES-GCM
    - AES-CCM

EF CORE 3.0

- Reworked LINQ
  - Faster, Better queries
- Exception for client-side query evaluation
- `dotnet ef: dotnet tool install --global dotnet-ef`
- Cosmos DB Support
- Interception of database operations
  - Before or After they happen
- EF 6.3 runs on .NET Core 3.0
- Nullability support
- Lazy Loading from 2.1, but this is the Devil

ASP.NET CORE 3.0

- Blazor
  - C# on the client side, Render HTML from C#
- gRPC
  - Contract based, Protobuf over HTTP/2, Streaming
- SignalR Updates
  - Autoreconnect, Custom authorization
- Identity Server 4
- Certificate Authorization
- IIS exceptions written to Windows Event Log

C# 8 STUFF

IntelliTect

DEFAULT INTERFACE IMPLEMENTATION

- Why?
- Fewer contract breakages
- Java and iOS compatibility
- Multiple inheritance
- Only .NET Core not Full Framework

```
interface IDeveloper  {
    void LearnNewLanguage(string language, DateTime dueDate);
    void LearnNewLanguage(string language) {
        // default implementation
       LearnNewLanguage(language, DateTime.Now.AddMonths(6));
    }
}
```

SWITCH STATEMENT UPGRADES (FAT ARROWS)

```
public static RGBColor FromRainbow(Rainbow colorBand) =>
    colorBand switch {
        Rainbow.Red => new RGBColor(0xFF, 0x00, 0x00),
        Rainbow.Orange => new RGBColor(0xFF, 0x7F, 0x00),
        Rainbow.Yellow => new RGBColor(0xFF, 0xFF, 0x00),
        Rainbow.Green => new RGBColor(0x00, 0xFF, 0x00),
        Rainbow.Blue => new RGBColor(0x00, 0x00, 0xFF),
        Rainbow.Indigo => new RGBColor(0x4B, 0x00, 0x82),
        Rainbow.Violet => new RGBColor(0x94, 0x00, 0xD3),
        _ => throw new ArgumentException(
            message: "invalid enum value", paramName: nameof(colorBand)),
    };
```

SWITCH STATEMENT UPGRADES (PROPERTIES)

```
public static decimal ComputeSalesTax(Address location, decimal salePrice) =>
        location switch {
                { State: "WA" } => salePrice * 0.06M,
                { State: "MN" } => salePrice * 0.75M,
                { State: "MI" } => salePrice * 0.05M,
                // ...
                _ => 0M
        };
```

SWITCH STATEMENT UPGRADES (TUPLES)

```
public static string RockPaperScissors(string first, string second) =>
        (first, second) switch {
                ("rock", "paper") => "rock is covered by paper. Paper wins.",
                ("rock", "scissors") => "rock breaks scissors. Rock wins.",
                ("paper", "rock") => "paper covers rock. Paper wins.",
                ("paper", "scissors") => "paper is cut by scissors. Scissors wins.",
                ("scissors", "rock") => "scissors is broken by rock. Rock wins.",
                ("scissors", "paper") => "scissors cuts paper. Scissors wins.",
                (_, _) => "tie"
        };
```

NULLABLE TYPE

- Null Reference Exception
- Object not set to an instance of an object
- <nullable>enable</nullable>
- Not nullable by default
- Implicitly nullable
- Dereferencing issues
  - Use null!
  - Turn off null checking for the file

VS2019 Launch Slides, Sample Code, …

IntelliTect.com/VS2019Launch