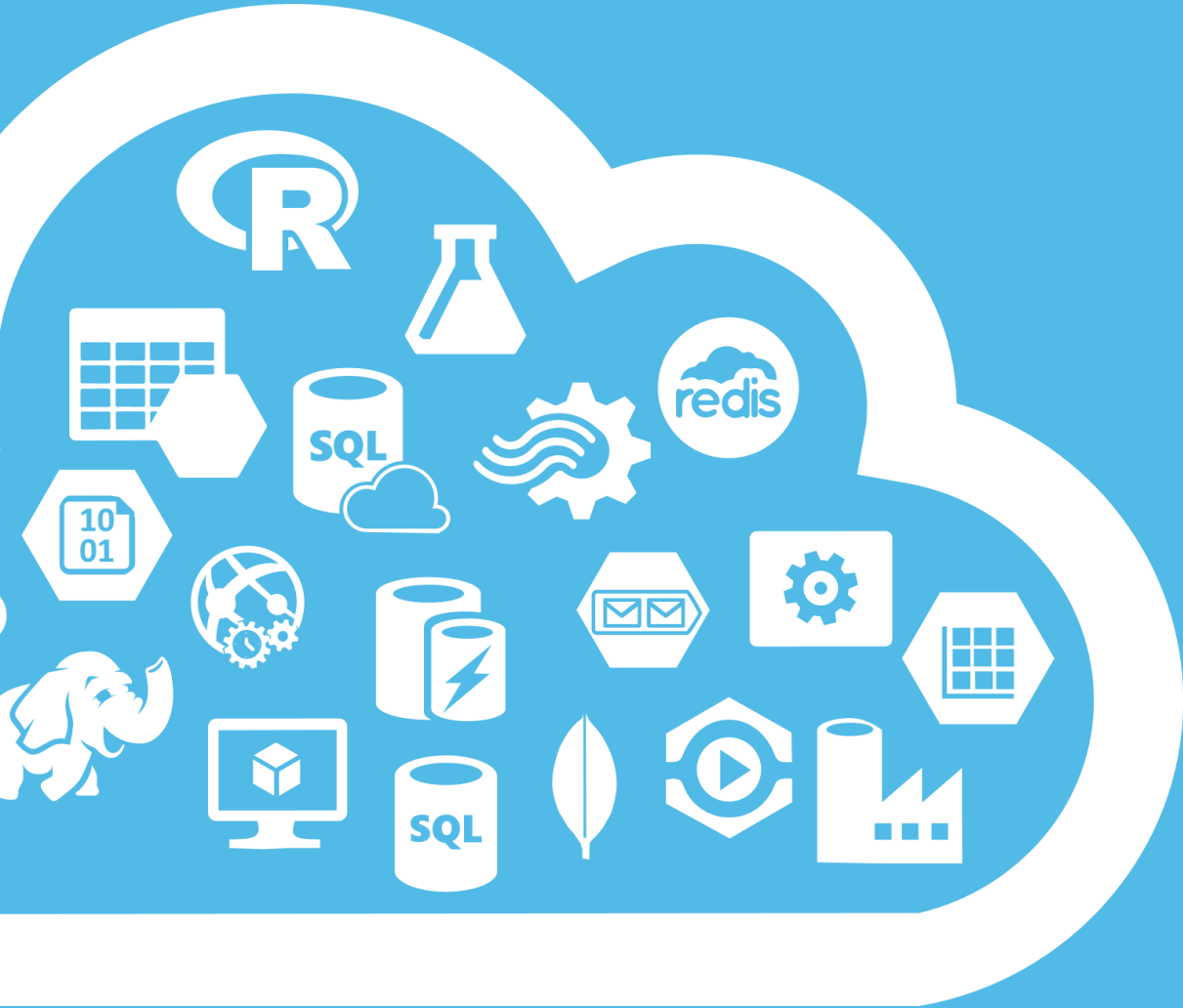


# Choosing .NET Technologies

August 2017



**Microsoft Partner**  
Gold Cloud Platform

**endjin**  
work smarter

# Navigating the ever-expanding world of .NET runtimes



Are you building a library, or an executable?

Library

choose a NetStandard2.0 library

Executable

it depends...

Where are you deploying your executable?

In an Azure App Service,  
Azure Functions,  
or Azure Batch (Linux or Windows)  
On Windows + MacOS + Linux

NetCoreApp 2.0

Where are you deploying your executable?

On a Windows Desktop machine

On a Windows Server / Service

choose Net461 in new-style projects

What if I depend on a 3<sup>rd</sup> Party Net461 Library?

Build a NetStandard2.0 library to  
expose the required subset of  
functionality

Reference the Net461 library from  
the NetStandard2.0 library

then...

What if I depend on a 3<sup>rd</sup> Party Net461 Library?

Write comprehensive tests in  
your target runtime to  
check for unsupported methods  
(Net461 or NetCoreApp2.0)

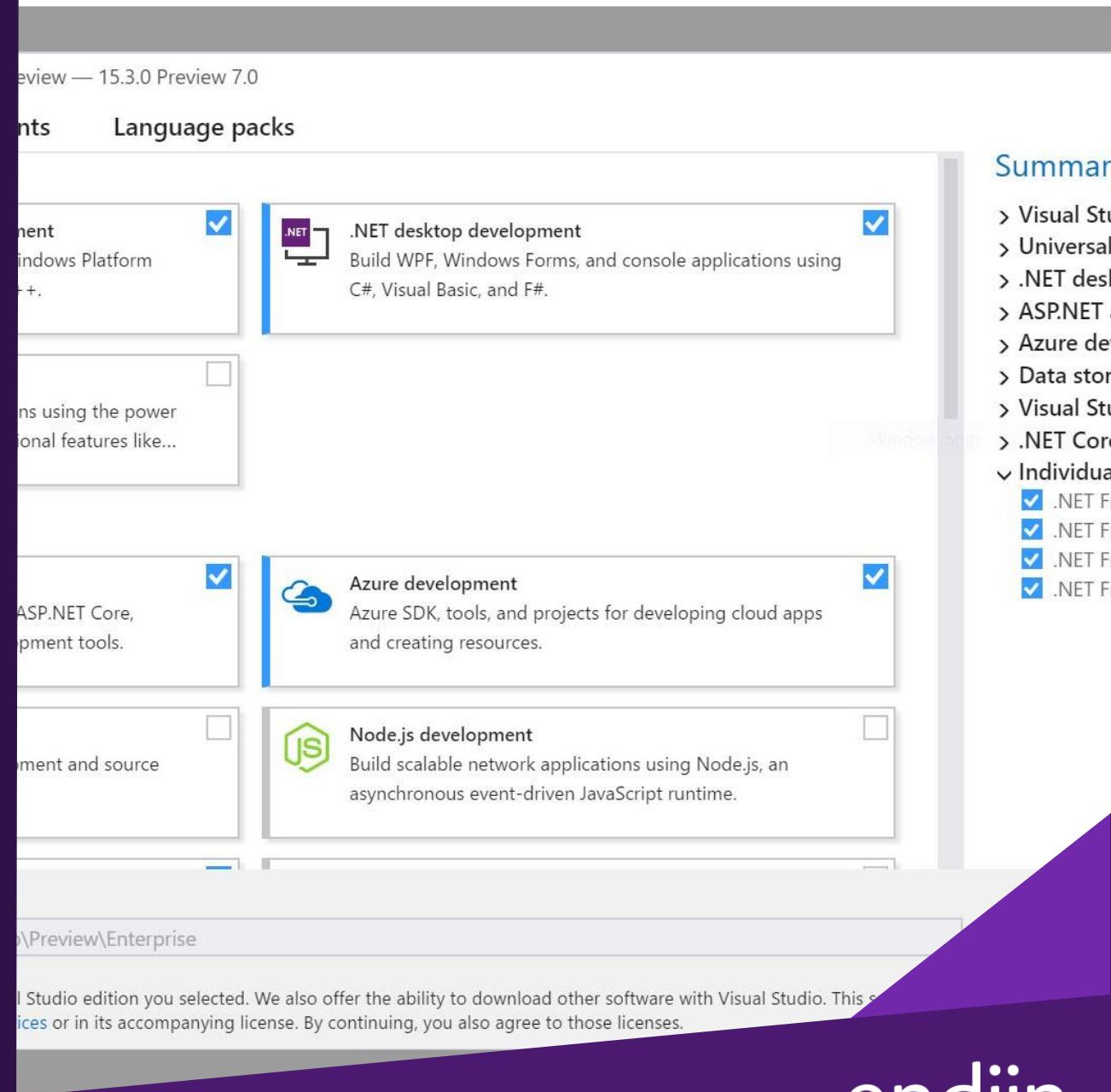
# What tools do I need?

Visual Studio 2017 15.3

.NET Core 2.0 SDK

VSTS hosted or private  
build agent

(things like code coverage and live  
tests work with this toolset)





# Common Scenarios

What if I'm building ASP.NET?

Build NetStandard2.0 libraries

Use AspNetCore2.0

Deploy using NetCoreApp2.0

What if I'm building WPF or Windows.Forms?

Build NetStandard2.0 libraries for  
common code

Build Net461 libraries for UI code

Build Net461 for Executable

What if I'm building a UWP app?

Keep doing whatever  
you are doing today

NetStandard2.0 support  
for UWP is coming

# What if I'm building F#?

Keep doing whatever  
you are doing today

Support for the new  
project system is coming



**Microsoft Partner**  
Gold Cloud Platform

**endjin**  
work smarter