

Building Ice Applications for .NET

This page provides important information for .NET developers.

On this page:

- [Building Ice Applications for .NET with Visual Studio](#)
- [Building Ice Applications for .NET with the .NET Core SDK](#)
- [Programming Language](#)
- [zeroc.ice.net NuGet Package](#)
 - [.NET Framework and .NET Standard Assemblies](#)
 - [Compression with bzip2](#)
- [Using the Sample Programs](#)

Building Ice Applications for .NET with Visual Studio

Install the following software and then refer to the [Ice Builder for Visual Studio instructions](#):

1. [A supported version of Visual Studio](#)

With Visual Studio 2017, you can optionally install the .NET Core cross-development toolset to create applications for .NET Core 2.0.

2. The [Ice Builder for Visual Studio](#) extension
3. The `zeroc.ice.net` NuGet package, described later on this page

[Back to Top ^](#)

Building Ice Applications for .NET with the .NET Core SDK

Install the following software and then refer to the [Ice Builder for MSBuild instructions](#):

1. The [.NET Core 2.0 SDK](#) for your operating system
2. The [zeroc.ice.net](#) NuGet package, described later on this page
3. The `slice2cs` compiler

`slice2cs` is a command-line tool written in C++ and available on most platforms

Platform	Distribution	Package with <code>slice2cs</code>
Ubuntu	apt packages	<code>zeroc-ice-compilers</code> ⚠ You need to install zeroc-ice-compilers from the 3.7.1 beta repo, as it includes a fix to the generated code for optional. The corresponding bug affects only Linux applications.
RHEL	RPMS	<code>ice-compilers</code> ⚠ You need to install ice-compilers from the 3.7.1 beta repo, as it includes a fix to the generated code for optional. The corresponding bug affects only Linux applications.
Windows	NuGet	<code>zeroc.ice.net</code>



On Windows, you can use [Ice Builder for Visual Studio](#) to configure Ice Builder for MSBuild, and the resulting projects can be used on any platform.

[Back to Top ^](#)

Programming Language

You can use any .NET programming language with Ice, however, the preferred programming language for Ice .NET applications is C# since:

- the only Slice language mapping for .NET is [Slice to C#](#)
- the only Slice compiler for .NET, `slice2cs`, generates C# code
- Ice for .NET is itself written in C#

[Back to Top ^](#)

zeroc.ice.net NuGet Package



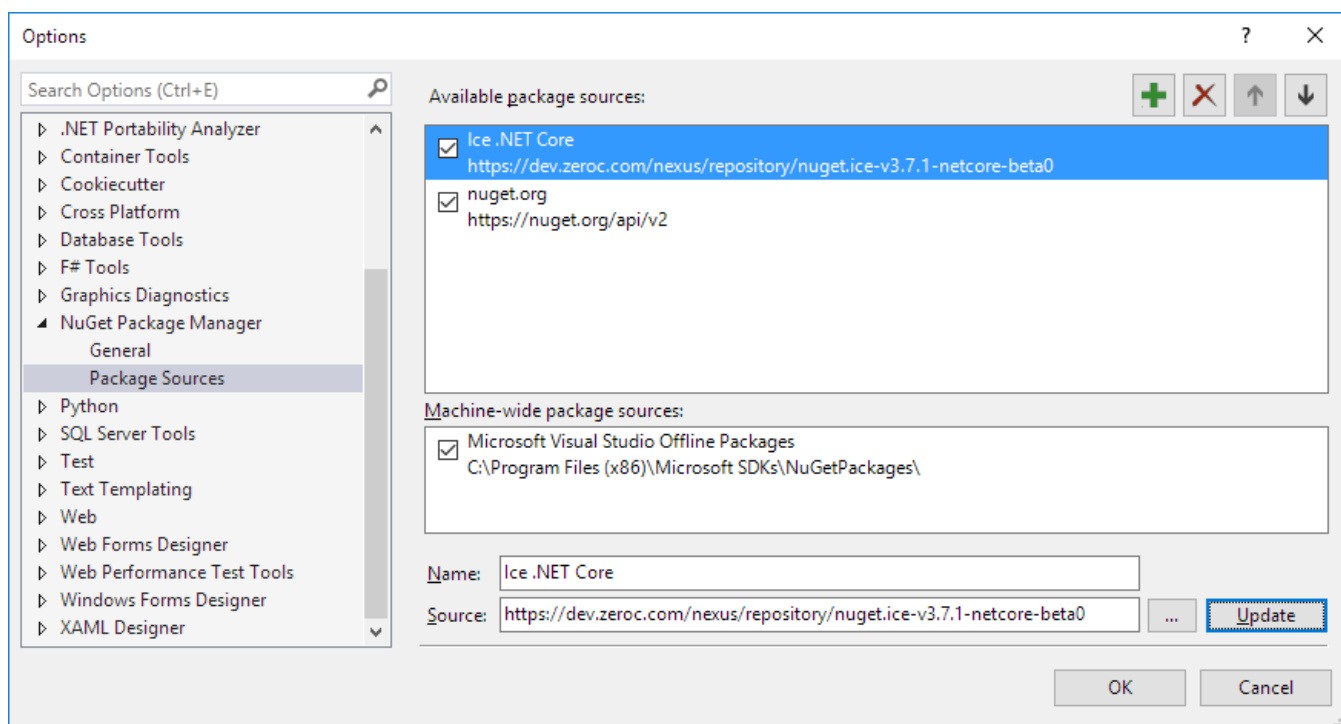
Upgraded on February 9, 2018 to version 3.7.1-beta1, compatible with the latest Ice Builder.

The Ice for .NET (zeroc.ice.net) NuGet package is organized as follows:

Folder	Contents
lib\net45	Assemblies for .NET Framework 4.5.1
lib\netstandard2.0	Assemblies for .NET Standard 2.0
tools	slice2cs.exe, slice2html.exe (Windows-only native tools)
tools\net45	iceboxnet.exe app for .NET Framework 4.5.1, bzip2.dll Windows x64 native library
tools\netcoreapp2.0	iceboxnet.dll app for .NET Core 2.0, bzip2.dll Windows x64 native library
build	MSBuild support files
slice	Slice files

For the 3.7.1 beta release, zeroc.ice.net is available from the beta repository <https://dev.zeroc.com/nexus/repository/nuget.ice-v3.7.1-netcore-beta0>.

On Windows, you can add this repository as a new package source using Visual Studio's "Tools > NuGet Package Manager > Package Manager Settings...":



On Linux you need to edit NuGet configuration (~/.nuget/NuGet/NuGet.Config) to add a new package source

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <packageSources>
    <add key="zeroc.com" value="https://dev.zeroc.com/nexus/repository/nuget.ice-v3.7.1-netcore-beta0/" />
    <add key="nuget.org" value="https://api.nuget.org/v3/index.json" protocolVersion="3" />
  </packageSources>
</configuration>
```

[Back to Top ^](#)

.NET Framework and .NET Standard Assemblies

zeroc.ice.net includes two sets of Ice assemblies: one set of assemblies for the .NET Framework 4.5 and another set for .NET Standard 2.0.

These assemblies are the same except for the differences described below:

	.NET Framework 4.5 Assemblies	.NET Standard 2.0 Assemblies
Run-time platform	Windows	Windows, Linux, macOS
Target Framework	.NET Framework 4.5.1 or greater on Windows	Any implementation of .NET Standard 2.0, including .NET Core 2.0 and .NET Framework 4.6.1.
Ice properties can be read from the Windows Registry	✓	✗
Signals caught by Ice. Application	Signal catching implemented using the Windows native function SetConsoleCtrlHandler .	Signal catching implemented using the portable .NET event Console.KeyPress .



The .NET Standard 2.0 assemblies are expected to work with [any .NET implementation of .NET Standard 2.0](#), however, they are currently tested and supported only with .NET Core 2.0 on Windows and Linux.

[Back to Top ^](#)

Compression with bzip2

Ice for .NET supports the optional compression of Ice requests and responses using the bzip2 native library. The bzip2 native DLL for Windows x64 is included in the zeroc.ice.net package. You can use the bzip2 system library on Linux and macOS.

[Back to Top ^](#)

Using the Sample Programs

Sample programs are available at the [ice-demos GitHub repository](#). You can browse this repository to see build and usage instructions for all supported programming languages. You can clone this repository with:

```
git clone -b 3.7.1-rc https://github.com/zeroc-ice/ice-demos.git
cd ice-demos
```

[Back to Top ^](#)