

Using the Sample Programs on Windows

The Ice installer for Windows optionally allows you to install a collection of sample programs in the directory of your choosing. Sample programs are provided for C++, Java, Android, C#, Visual Basic, Ruby, Python and PHP. Below you will find build instructions for languages, compilers, and target platforms. These instructions assume that you have configured your system according to the instructions in the [release notes](#).

On this page:

- [Building and running the C++ demos \(Visual Studio 2010SP1, 2012, or 2013\)](#)
- [Building and running the C++ demos \(Visual C++ Express 2010SP1, 2012, or 2013\)](#)
- [Building and running the C++ WinRT demos \(Visual Studio 2012 or 2013\)](#)
- [Building and running the .NET demos](#)
 - [Building the C# demos](#)
 - [Building the Visual Basic demos](#)
 - [Running the .NET demos](#)
 - [SSL notes for the .NET demos](#)
- [Building and running the .NET Compact Framework demos \(Visual Studio 2008\)](#)
- [Building and running the Silverlight demos \(Visual Studio 2010SP1, 2012, or 2013\)](#)
- [Building and running the Java demos](#)
- [Building and running the Android demos](#)
- [Running the Python demos](#)
- [Running the Ruby demos](#)
- [Building and running the PHP demos](#)

Building and running the C++ demos (Visual Studio 2010SP1, 2012, or 2013)



The Visual Studio project files require the [Ice Visual Studio Add-In](#), which is installed with the Ice binary distribution.

The C++ demos are in the `demo` directory.

To build the C++ demos, start Visual Studio and open the solution `demo\demo.sln`. The demo projects were created with Visual Studio 2010 and will be converted if you are using a new version of Visual Studio.

Select your target configuration: Debug or Release, Win32 or x64 (on supported x64 platforms). Right click on the desired demo in the Solution Explorer window and select "Build".

To run a demo, open a command window and change to the desired demo subdirectory. Review the `README` file if one is present. Type `server` to start the server. In another command window, type `client` to start the client.

This distribution also includes a few demos for integrating Ice with Oracle. Open the solution `demo\demo-oracle.sln` in Visual Studio 2010SP1 to build the corresponding projects.

Building and running the C++ demos (Visual C++ Express 2010SP1, 2012, or 2013)



Visual C++ Express does not support the [Ice Visual Studio Add-In](#), therefore you must use `NMAKE` to build the C++ demos.

The C++ demos are in the `demo` directory.


To build the C++ demos, open a "Visual Studio Command Prompt" window. If you installed Ice in a non-default location, set `ICE_HOME` as shown below:

```
> set ICE_HOME=<Ice installation root directory>
```

Change to the `demo` directory and run the following command to build the demos:

```
> nmake /f Makefile.mak
```

To run a demo, change to the desired demo subdirectory and review the `README` file if one is present. Type `server` to start the server. In another command window, type `client` to start the client.

 Two of the Ice demos require the Microsoft Foundation Classes (MFC). These demos will not compile because the Express edition does not include MFC support. The demos are located in `demo\Ice\MFC` and `demo\IcePatch2\MFC`.

Building and running the C++ WinRT demos (Visual Studio 2012 or 2013)

 The Visual Studio project files require the [Ice Visual Studio Add-In](#), which is installed with the Ice binary distribution.

The C++ demos are in the `demo` directory.

WinRT requires Windows 8 with Visual Studio 2012, or Windows 8.1 with Visual Studio 2013. You can choose from two solutions:

- `demo\demo-winrt.sln` - Targets Windows 8, supports Visual Studio 2012 and 2013
- `demo\demo-winrt-8.1.sln` - Targets Windows 8.1, supports Visual Studio 2013 only

To build the C++ WinRT demos, start Visual Studio and open the desired solution.

Select your target configuration: Debug or Release, Win32 or x64 (on supported x64 platforms). Right click on the desired demo in the Solution Explorer window and select "Build".

To run a demo, review the `README` file in the desired demo subdirectory.

To run the demos with a device such as Surface, you will also need to re-create the server certificate to try out SSL. The Windows RT SSL implementation requires the certificate's common name to match the IP address or DNS name of the target server. We provide a `makewinrtcerts.py` Python script to re-generate the server certificate in the `certs` directory at the root of the Ice demo distribution. To run the script:

```
C:\Users\acme\Documents\ZeroC\Ice-3.5.1-demos> cd certs
C:\Users\acme\Documents\ZeroC\Ice-3.5.1-demos\certs> makewinrtcerts.py 192.168.1.50
```

In the example above, the server certificate's common name will be set to `192.168.1.50`. The certificate is generated in the `certs\winrt` directory. You should specify the IP address of the machine running the SSL server here.

Building and running the .NET demos

Sample programs are provided for C# and Visual Basic.

Building the C# demos

 The Visual Studio project files require the [Ice Visual Studio Add-In](#), which is installed with the Ice binary distribution.

The C# demos are in the `democs` directory.

To build a C# demo, start Visual Studio and open the solution `democs\demo.sln`. Right click on the desired demo in the Solution Explorer window and select "Build".

Building the Visual Basic demos

 The Visual Studio project files require the [Ice Visual Studio Add-In](#), which is installed with the Ice binary distribution.

The Visual Basic demos are in the `demovb` directory.

To build a Visual Basic demo, start Visual Studio and open the solution `demovb\demo.sln`. Right click on the desired demo in the Solution Explorer window and select "Build".

Running the .NET demos

To run a demo, open a command window, change to the desired demo subdirectory and review the `README` file if one is present. Type `server` to start the server. In another command window, type `client` to start the client.


SSL notes for the .NET demos

In order to use SSL with the sample programs, an SSL certificate must be installed on your system. The configuration files handle this for you, but you will be presented with a confirmation dialog the first time you run a sample program that uses SSL.

Once you are finished with the sample programs, follow these steps to remove the certificate:

1. Start Internet Explorer.
2. Select Internet Options from the Tools menu.
3. Select the Content tab and click the "Certificates" button.
4. Select the Trusted Root Certification Authorities tab.
5. Select the entry for "ZeroC Test CA", click the Remove button, and confirm that you want to remove this certificate.

Building and running the .NET Compact Framework demos (Visual Studio 2008)


 The Visual Studio project files require the [Ice Visual Studio Add-In](#), which is installed with the Ice binary distribution.

The demos for the .NET Compact Framework are located in the `democs\Ice\compact` subdirectory.

To build the demo, start Visual Studio 2008 and open the solution `democs\democf.sln`. Right click on the demo in the Solution Explorer window and select "Build".

Review the `README.txt` file in the demo project for instructions on starting a device emulator and deploying the program.

Building and running the Silverlight demos (Visual Studio 2010SP1, 2012, or 2013)

 The Visual Studio project files require the [Ice Visual Studio Add-In](#), which is installed with the Ice binary distribution.

The demos for Silverlight are located in `democs\Ice\sl` and `democs\Glacier2\sl` subdirectories.

To build the demo, start Visual Studio and open the solution `democs\demo1.sln`. Right click on the demo in the Solution Explorer window and select "Build".

To run a demo review the `README.txt` file in the demo project.

Building and running the Java demos

The Java demos are in the `demoj` directory.

To build the Java demos, you need J2SE SDK 1.6.0 or later, and Ant 1.7.0 or later. They can be downloaded from:

<http://www.oracle.com/technetwork/java/javase/overview/index.html>
<http://ant.apache.org/bindownload.cgi>

The `ant bin` directory must be added to your `PATH`, and the following environment variables need to be defined:

```
> set JAVA_HOME=<Java SDK installation root directory>
> set PATH=<Ice installation root directory>\bin;%PATH%
```

If you installed Ice in a non-default location, set `ICE_HOME` as shown below:

```
> set ICE_HOME=<Ice installation root directory>
```

You can build all of the demos by running `ant` in the `demoj` directory, or you can build individual demos by changing to the desired subdirectory and then running `ant`.

Before running the demos you must modify your `CLASSPATH` as follows:

```
> set CLASSPATH=classes;%CLASSPATH%
```

Your `CLASSPATH` should also include `Ice.jar`, `Freeze.jar`, and other service-specific JAR files, as per the [Release Notes](#) instructions.

To run a demo, open a command window, change to the desired demo directory, and type `java Server` to run the server. In a separate command window, type `java Client` to run the client.

Some demo directories contain `README` files if additional requirements are necessary.

Building and running the Android demos

Several sample Android projects are provided in the `demoj/android` subdirectory. You must use Eclipse and the [Slice2Java plug-in](#) to build these projects.

In Eclipse, you can open a sample project by choosing **File->Import...**; in the "General" group, select "Existing Project into Workspace", then open one of the subdirectories of `demoj/android`.

The sample projects are configured to locate the Ice run time JAR file (`Ice.jar`) via the [ICE_HOME classpath variable](#).

If you installed `Ice.jar` in a different location, you will need to add it as an external JAR file in each sample project:

1. Open the project's properties and select Java Build Path
2. Click on the Libraries tab
3. Click Add External JARs... and navigate to `Ice.jar`
4. Click OK to save your settings

After successfully building an Android project, deploy it onto a suitable emulator or device and review the `README` file in the project subdirectory for further instructions.

Running the Python demos

The Python demos are in the `demopy` directory.

You need Python 3.3 to run the demos. A binary installer for Python can be downloaded from:

<http://www.python.org/download>

To run a demo, open a command window, change to the desired demo directory, and type `python Server.py` to start the server. In a separate command window, type `python Client.py` to run the client.

Running the Ruby demos

The Ruby demos are in the `demorb` directory.

You need Ruby 1.9.3 to run the demos. A binary installer for Ruby can be downloaded from:

<http://rubyinstaller.org/downloads/>

Since Ice for Ruby does not support server-side activities, only clients are provided in the demos. In order to run the demos you must use the corresponding C++ server.

For example, to run the hello application in `demo\Ice\hello`, we begin by starting the C++ server:

```
> cd <Ice demo root directory>\demo\Ice\hello
> server
```

Then in a separate command window, start the Ruby client:

```
> cd <Ice demo root directory>\demorb\Ice\hello  
> ruby Client.rb
```

Building and running the PHP demos

PHP demos are provided in the `demophp` directory.

You will need PHP 5.4 to run the demos. It can be downloaded from:

<http://www.php.net/downloads.php>

Building the demos requires Microsoft NMAKE. If you do not already have NMAKE installed, you can download it as part of the Windows Platform SDK:

<http://www.microsoft.com/downloads/details.aspx?FamilyId=E6E1C3DF-A74F-4207-8586-711EBE331CDC&displaylang=en>

Open a command window. If you installed Ice in a non-default location, set `ICE_HOME` as shown below:

```
> set ICE_HOME=<Ice installation root directory>
```

Change to the `demophp` directory and run the following command to build the demos:

```
> nmake /f Makefile.mak
```

The examples in `demophp\Ice\hello` and `demophp\Glacier2\hello` demonstrate using the Ice extension for PHP in a dynamic Web page, whereas the example in `demophp\Ice\value` requires PHP's command-line interpreter. A `README` file is provided in each of the example directories.