

Using the Sample Programs

Sample programs are [available for download](#) for C++, Java, Android, C#, Visual Basic, Ruby, Python and PHP. Below you will find instructions for building and running these programs. The instructions assume that you have configured your platform according to the instructions in the [release notes](#).

On this page:

- [Building the Demos](#)
 - [Demo Prerequisites](#)
 - [Demo Build Instructions](#)
 - [Demo Build Instructions for Android](#)
- [Running the Ice Demos](#)
 - [Running the C++ demos](#)
 - [Running the Java demos](#)
 - [Running the Android demos](#)
 - [Running the C# demos](#)
 - [Running the Python demos](#)
 - [Running the Ruby demos](#)
 - [Running the PHP demos](#)

Building the Demos

This section describes how to build the demos included in the Ice demo distribution.

The demo archive contains sample programs for each language mapping that Ice supports. Not all demos will build against all Ice distributions. Distributions for specific platform/tool combinations may not support every Ice language mapping.

Depending on the language mapping, various build tools are required for building the demos. The C++, C#, and PHP demos use makefiles, the Java demos use the Apache Foundation's ant build tool, and the Android demos require Eclipse and the [Slice2Java plug-in](#).



The Ruby and Python demos do not need to be built.

Demo Prerequisites

The makefiles for C++, C#, and PHP require GNU make 3.80 or later. If your system does not come with GNU make, you can find information on how to acquire it at the following link:

<http://www.gnu.org/software/make/>

The Ice for Java demos require Ant 1.7.0, but we recommend using the most recent release available. Ant can be obtained at:

<http://ant.apache.org/bindownload.cgi>

The Ice for Android demos require Eclipse and the [Slice2Java plug-in](#). The demos were tested with Eclipse Helios (3.6).

<http://www.zeroc.com/eclipse.html>

Demo Build Instructions

To build the demos you need to:

- Configure your system according to the directions in the [release notes](#) (for binary distributions) or the platform-specific [build instructions](#) (if you built Ice from sources).
- If you are using a non-RPM installation and you did not install Ice in its default location (`/opt/Ice-3.5.1`), set the environment variable `ICE_HOME` to point to your Ice installation root directory. For example, in a Bash shell:

```
$ export ICE_HOME=$HOME/testing/Ice-3.5.1
```

With a non-RPM installation, you also need to either create an `/opt/Ice-3.5` symbolic link (as described in the [release notes](#)), or add the Ice library directory to your shared library search path. For example, on Linux x86:

```
$ export LD_LIBRARY_PATH=$ICE_HOME/lib:$LD_LIBRARY_PATH
```

The Ice library directory is `$ICE_HOME/lib64` on Linux x86_64 and `$ICE_HOME/lib/sparcv9` on Solaris for SPARC 64-bit.

- Review the build settings found in `config/Make.rules` (C++), `config/Make.rules.cs` (C#), `config/Make.rules.php` (PHP) and `config/build.properties` (Java) and adjust any you want changed. For example, for C++ and C#, set `OPTIMIZE=yes` in order to build with optimization, or for C++ set `LP64=yes` if you are building against a 64-bit Ice installation (on OS X, set `CXXARCHFLAGS` to build 32-bit and 64-bit FAT binaries).
- Build the demos. For example:

```
# C++ demos
$ cd Ice-3.5.1-demos/demo
$ make
# C# demos
$ cd Ice-3.5.1-demos/demoscs
$ make
# PHP demos
$ cd Ice-3.5.1-demos/demophp
$ make
# Java demos
$ cd Ice-3.5.1-demos/demoj
$ ant
```

Demo Build Instructions for Android

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_JAR_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

Running the Ice Demos

Most demos consist of a single server and client. The instructions to run a demo depend on the programming language. Instructions for each language are included below. Some demo directories contain a `README` file if additional steps are necessary.

Running the C++ demos

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

To run a demo, open a terminal window, change to the desired demo directory, and enter the following command to run the server:

```
$ ./server
```

To run the client, open another terminal window, change to the same demo directory, and run:

```
$ ./client
```

Running the Java demos

Java demos are provided in the `demoj` directory.

After a successful build, the compiled classes are stored in a subdirectory named `classes` in each demo directory. You need to add this subdirectory to your `CLASSPATH`. For example, with a bash shell:

```
$ export 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 terminal window, change to the desired demo directory, and enter the following command to run the server:

```
$ java Server
```

To run the client, open another terminal window, change to the same demo directory, and run:

```
$ java Client
```

Running the Android demos

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 C# demos

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

The C# demos are only supported on SuSE Enterprise Linux Server and Ubuntu but may work on other platforms. Note however that the Ice for .NET run time is included only in the binary distributions for SuSE Enterprise Linux Server and Ubuntu.

To run a demo, open a terminal window, change to the desired demo directory, and enter the following command to run the server:

```
$ mono server.exe
```

To run the client, open another terminal window, change to the same demo directory, and run:

```
$ mono client.exe
```

Running the Python demos

Python demos are provided in the `demopy` directory.

To run a demo, open a terminal window, change to the desired demo directory, and enter the following command to run the server:

```
$ python Server.py
```

To run the client, open another terminal window, change to the same demo directory, and run:

```
$ python Client.py
```

Running the Ruby demos

Ruby demos are provided in the `demorb` directory.

Ice provides only client demos in Ruby since Ice for Ruby does not support server-side activities. In order to run a sample client, you must first start its corresponding server from another Ice language mapping. For example, start the C++ hello server:

```
$ cd $HOME/testing/Ice-3.5.1-demos/demo/Ice/hello
$ ./server
```

To run the client, open another terminal window, change to the corresponding demo directory, and run:

```
$ cd $HOME/testing/Ice-3.5.1-demos/demorb/Ice/hello
$ ruby Client.rb
```

Running the PHP demos

PHP demos are provided in the `demophp` directory.

The Ice extension for PHP is provided as a dynamically-loadable shared library in Linux RPM distributions. On all other Unix platforms, you will need to build PHP and the Ice extension from source code. Please refer to the `php/INSTALL` file included in the Ice source distribution for more information.

The examples in `demophp/Ice/hello` and `demophp/Glacier2/hello` demonstrate the use of the Ice extension for PHP in a dynamic Web page, whereas the example in `demophp/Ice/value` requires PHP's command line interpreter. All examples require that an Ice server be available; a matching server from any of the other language mappings can be used. A `README` file is provided in each of the example directories.

Note that you may need to modify the `php.ini` files in each demo directory to match your PHP installation and ensure that the Ice extension is loaded properly.