Building Ice Applications in Java

This page provides important information for Java developers.

On this page:

Prerequisites

In order to build applications with Ice in Java, you need:

1. the Ice JAR files (`ice.jar`, `icediscovery.jar`, `icegrid.jar`, etc.)
   These JAR files are provided through Maven, as described below. Ice binary distributions for Debian and Ubuntu also include these JAR files.

2. the `slice2java` compiler
   `slice2java` is a command-line tool written in C++ and available on most platforms

Platform	Distribution	Package with slice2java
Debian and Ubuntu	apt packages	zeroc-ice-compilers
RHEL, SLES, Amazon Linux	RPMs	ice-compilers
macOS	homebrew	always installed
Windows	MSI	always installed
Windows	NuGet	zeroc.ice.v100, zeroc.ice.v120, zeroc.ice.v140, zeroc.ice.v141

3. the Ice Builder for Gradle, if you are using Gradle.

Back to Top ^

Java and Java Compat Mappings

Ice provides two distinct Slice-to-Java mappings:

- Java
  This is a new mapping that takes advantage of features in Java 8. We recommend you select this mapping for new Ice-based applications written in Java.

- Java Compat
  This mapping is largely backward-compatible with prior Ice releases. Although Ice 3.7 no longer supports Java versions prior to Java 8, the "Compat" mapping does not depend on any Java 8-specific language or run-time features.

slice2java, the Slice-to-Java code generator, generates code for the Java mapping by default.

Back to Top ^

Maven Repository

You can fetch all Ice build artifacts from the Maven Central repository. ZeroC provides the following JAR files, all in group `com.zeroc` with version `3.7.0`:

Java

Name	Description
`glacier2`	Generated proxy and skeleton classes plus helper classes for connecting to the Glacier2 service
`ice`	Ice core
`icebox`	The IceBox server, and generated proxy and skeleton classes for connecting to IceBox
`icebt`	The IceBT plug-in (only for Android)

`icediscovery`	The IceDiscovery plug-in
`icegrid`	Generated proxy and skeleton classes for connecting to the IceGrid service
`icelocatordiscovery`	The IceLocatorDiscovery plug-in
`icepatch2`	Generated proxy and skeleton classes for connecting to the IcePatch2 service
`icessl`	The IceSSL plug-in
`icestorm`	Generated proxy and skeleton classes for connecting to the IceStorm service

Java Compat

Name	Description
`glacier2-compat`	Generated proxy and skeleton classes plus helper classes for connecting to the Glacier2 service
`ice-compat`	Ice core and IceSSL plug-in
`icebox-compat`	The IceBox server, and generated proxy and skeleton classes for connecting to IceBox
`icebt-compat`	The IceBT plug-in (only for Android)
`icediscovery-compat`	The IceDiscovery plug-in
`icegrid-compat`	Generated proxy and skeleton classes for connecting to the IceGrid service
`icelocatordiscovery-compat`	The IceLocatorDiscovery plug-in
`icepatch2-compat`	Generated proxy and skeleton classes for connecting to the IcePatch2 service
`icestorm-compat`	Generated proxy and skeleton classes for connecting to the IceStorm service

Back to Top ^

Bzip2

Ice for Java supports protocol compression using the BZip2 implementation included with Apache Commons Compress. Compression is automatically enabled if these classes are present in your `CLASSPATH`. The Maven package id for the JAR file is as follows:

groupId	version	artifactId
org.apache.commons	1.14	commons-compress

> These classes are a pure Java implementation of the bzip2 algorithm and therefore add significant latency to Ice requests.

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 https://github.com/zeroc-ice/ice-demos.git
cd ice-demos
```

Back to Top ^