Using the macOS Binary Distribution

This page provides important information for users of the Ice binary distribution for macOS.

On this page:

Overview of the macOS binary distribution

Ice for C++, Java, Objective-C, and PHP on macOS are provided through Homebrew. The `ice@3.6` formula includes the following components by default:

- Run-time libraries for C++, Objective-C and PHP
- Run-time libraries for the Freeze service in C++
- Executables for IceGrid, IceStorm, Glacier2, and IcePatch2 services
- Tools and libraries for developing Ice applications

More information about Homebrew is available at http://brew.sh.

> By default, the `ice@3.6` formula doesn't build any Java component - this includes Ice for Java, the Freeze service for Java, and the IceGrid Admin graphical tool.

Installing the macOS binary distribution

Using Homebrew, you can install the distribution with this command:

```
brew install zeroc-ice/tap/ice@3.6 [--with-java]
```

The `--with-java` option builds the Java components and the IceGrid Admin app. You can also install IceGrid Admin on its own by downloading Ice Grid Admin.dmg.

Setting up your macOS environment to use Ice

After installing Ice, read the relevant language-specific sections below to learn how to configure your environment and start programming with Ice.

C++

The distribution includes libraries built with LLVM `libc++`. These libraries also support C++11 applications when compiled with the `--std=c++11` option.

When compiling Ice for C++ programs, you must pass the `-pthread` option. A typical compile command would look like this:

```
c++ -c -pthread myprogram.cpp
```

When linking a program you must link with at least `libIce` and `libIceUtil`. A typical link command would look like this:

```
c++ -o myprogram myprogram.o -lIce -lIceUtil
```

Additional libraries are necessary if you are using an Ice service such as IceGrid or Glacier2.

If you want to include Ice in your application bundle, you will need to copy the necessary Ice libraries to the `Contents/Frameworks` subdirectory of your bundle and use `@loader_path/../Frameworks` as the run path when linking the application.

Please refer to the `dyld` man page on your macOS system to learn more about `@loader_path`.

Objective-C

When compiling Ice for Objective-C programs, you must pass the `-pthread` option. A typical compile command would look like this:

```
cc -c -pthread myprogram.m
```

When linking a program you must link with `libIceObjC`. A typical link command would look like this:

```
cc -o myprogram myprogram.o -lIceObjC -framework Foundation
```

Additional libraries are necessary if you are using an Ice service such as IceGrid or Glacier2.

If you want to include Ice in your application bundle, you will need to copy the necessary Ice libraries to the `Contents/Frameworks` subdirectory of your bundle and use `@loader_path/../Frameworks` as the run path when linking the application.

Please refer to the `dyld` man page on your macOS system to learn more about `@loader_path`.

PHP

The macOS distribution includes the Ice extension for PHP and the Slice-to-PHP compiler (`slice2php`).

The PHP interpreter loads the Ice extension automatically when you add the following directive to `/etc/php.ini`:

```
extension=/usr/local/php/extensions/IcePHP.dy
```

Additional configuration directives can also be added to `/etc/php.ini`.

You can verify that the Ice extension is installed properly by examining the output of the `php -m` command, or by calling the `phpinfo()` function from a script.

Your application will also need to include at least some of the Ice for PHP run-time source files (installed in `/usr/local/lib/share/php`). You can modify the `include_path` setting in `php.ini` to add the installation directory:

```
include_path = /usr/local/lib/share/php:...
```

Another option is to modify the include path from within your script prior to including any Ice run-time file:

PHP

```
ini_set('include_path', ini_get('include_path') . PATH_SEPARATOR . '/usr/local/opt/ice/lib/share/php');
require 'Ice.php'; // Load the core Ice run time definitions.
```

Berkeley DB utilities

Berkeley DB utilities are installed in`/usr/local/opt/ice/libexec/bin`. You may add this directory to your `PATH`, or call the executables directly.

Using the sample programs on macOS

Sample programs for all programming languages are available in a separate GitHub repository. Simply clone this repository and use its 3.6 branch:

```
git clone -b 3.6 https://github.com/zeroc-ice/ice-demos.git
cd ice-demos
```

Starting IceGrid Admin on macOS

You can launch IceGrid Admin with the `IceGridAdmin` application installed in your `/Applications` directory. This app is a Java program and requires JRE 7u6 or later.