

Using the Linux Binary Distributions

This page provides important information for users of the Ice binary distributions on Linux platforms.

On this page:

- [Overview of the Ice Binary Distributions for Linux](#)
 - [Ubuntu Packages](#)
 - [Debian Packages](#)
 - [RPM Packages](#)
 - [Bi-Arch Support](#)
- [Installing the Linux Packages](#)
 - [Installing Ice on Ubuntu](#)
 - [Installing Ice on Debian 9 \(stretch\)](#)
 - [Installing Ice on Red Hat Enterprise Linux 7](#)
 - [Installing Ice on Red Hat Enterprise Linux 6](#)
 - [Installing Ice on Amazon Linux](#)
 - [Installing Ice on SUSE Linux Enterprise Server 12](#)
 - [Installing Ice on SUSE Linux Enterprise Server 11](#)
- [Setting up your Linux environment to use Ice](#)
 - [C++](#)
 - [PHP](#)
- [Using the sample programs on Linux](#)
- [Starting IceGrid Admin on Linux](#)
- [Startup scripts for IceGrid and Glacier2 services](#)

Overview of the Ice Binary Distributions for Linux

Ubuntu Packages

ZeroC provides the following packages for Ubuntu:

Package	Description
<code>zeroc-ice-all-dev</code>	Meta package that installs all development packages
<code>zeroc-ice-all-runtime</code>	Meta package that installs all run-time packages, servers and utilities
<code>libzeroc-freeze3.6</code>	Freeze for C++ run-time libraries
<code>libzeroc-ice-dev</code>	C++ header files, libraries, and Slice compiler for C++
<code>libzeroc-ice-java</code>	Java development kit, including JAR files and Slice compiler for Java
<code>libzeroc-ice3.6</code>	C++ run-time libraries
<code>libzeroc-icestorm3.6</code>	IceStorm service for IceBox C++
<code>php5-zeroc-ice</code>	PHP extension and run time files
<code>php5-zeroc-ice-dev</code>	Slice compiler for PHP development
<code>zeroc-glacier2</code>	Glacier2 service
<code>zeroc-ice-slice</code>	Slice files
<code>zeroc-ice-utils</code>	Utilities necessary for administering an Ice installation
<code>zeroc-ice-utils-java</code>	Additional utilities that require Java, such as the IceGrid Admin GUI
<code>zeroc-icebox</code>	IceBox server for C++
<code>zeroc-icegrid</code>	IceGrid service
<code>zeroc-icepatch2</code>	IcePatch2 service

Debian Packages

ZeroC provides the following packages for Debian:

Package	Co-installable	Description
<i>zeroc-ice-all-dev</i>	no	Meta package that installs all development packages
<i>zeroc-ice-all-runtime</i>	no	Meta package that installs all run-time packages, servers and utilities
<i>libzeroc-freeze3.6</i>	yes	Freeze for C++ run-time libraries
<i>libzeroc-ice-dev</i>	yes	Ice for C++ libraries and header files
<i>libzeroc-ice3.6</i>	yes	C++ run-time libraries
<i>libzeroc-ice3.6-java</i> (Debian 9)	n/a	Ice for Java run-time libraries
<i>libzeroc-icestorm3.6</i>	yes	IceStorm publish-subscribe event distribution service
<i>php-zeroc-ice</i> (Debian 9)	no	PHP extension
<i>python3-zeroc-ice</i> (Debian 9)	no	Python 3 extension
<i>zeroc-glacier2</i>	no	Glacier2 service
<i>zeroc-ice-compilers</i>	no	Slice compilers for developing Ice applications
<i>zeroc-ice-slice</i>	n/a	Slice files
<i>zeroc-ice-utils</i>	no	Ice utilities and admin tools
<i>zeroc-icebox</i>	no	IceBox server for C++
<i>zeroc-icegrid</i> (Debian 9)	no	IceGrid service
<i>zeroc-icegridgui</i> (Debian 9)	n/a	IceGrid GUI admin tool
<i>zeroc-icepatch2</i> (Debian 9)	no	IcePatch2 service



A co-installable package is a package for which you can install several architectures (amd64, armhf...) at the same time.



The main difference between the Ubuntu and Debian packages is the location of the Slice compilers. With the Ubuntu packages, they are packaged alongside the other dev-kit files, for example `slice2cpp` is included in `libzeroc-ice-dev`. With the Debian packages, all the Slice compilers are in the `zeroc-ice-compilers` package.

RPM Packages

ZeroC provides the following RPMs for Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and Amazon Linux:

RPM	Description
<i>ice-all-devel</i>	Meta package that installs all development packages
<i>ice-all-runtime</i>	Meta package that installs all run-time packages, servers and utilities
<i>glacier2</i>	Glacier2 service
<i>ice-utils</i>	Utilities necessary for administering an Ice installation
<i>ice-utils-java</i>	Additional utilities that require Java, such as the IceGrid Admin GUI

ice-slice	Slice files
icebox	IceBox server for C++
icegrid	IceGrid service
icepatch2	IcePatch2 executable
libfreeze3.6-c++	Freeze for C++ run-time libraries
libice-c++-devel	C++ header files, libraries, and Slice compiler for C++
libice-java	Java development kit, including JAR files and Slice compiler for Java
libice3.6-c++	C++ run-time libraries
libicestorm3.6	IceStorm service
php-ice	PHP extension and run time files
php-ice-devel	Slice compiler for PHP development

ZeroC also supplies RPMs for the following third-party packages:

RPM	Description
db53	Berkeley DB 5.3.28 C and C++ run time libraries
db53-devel	C++ development files for Berkeley DB 5.3.28
db53-java	Berkeley DB 5.3.28 Java run time
db53-utils	Berkeley DB 5.3.28 command-line utilities
mcpp-devel	MCP C++ preprocessor library

The db53 packages are supplied only for Amazon Linux, Red Hat Enterprise Linux 6 and SUSE Linux Enterprise Server. The db53-devel and the mcpp-devel RPMs are only necessary for building Ice from source.

On Red Hat Enterprise Linux 7, Berkeley DB 5.3 is available as packages libdb-cxx, libdb-java and libdb-utils; some of these packages are in the `rhel-7-server-optional-rpms` repository that you may need to [enable on your system](#).

Bi-Arch Support

Recent Linux releases such as RHEL 7 are moving away from separate 64-bit and 32-bit distributions and migrating to a single 64-bit release that includes support for 32-bit packages. On these systems, all of the Ice packages listed above are provided for the x86_64 architecture, along with a limited subset of packages for the x86 architecture. The subset includes C++ run-time and development libraries, along with 32-bit versions of the [IceBox](#) server and [IceStorm](#) service. The 32-bit IceBox package installs the executables as `icebox32` and `icebox32++11`; the latter is used for IceBox services built in C++11 mode, as described in the C++ section below.

For development purposes, you will still need to install the 64-bit development kit packages: the 32-bit development kit packages complement these 64-bit packages.

On other systems, such as Amazon Linux and SLES 12, we only provide packages for the x86_64 architecture.

Installing the Linux Packages

This section describes how to install Ice binary packages for all of the supported Linux platforms.

Installing Ice on Ubuntu

Follow the instructions below to install Ice on Ubuntu.

1. Install ZeroC's key to avoid warnings with unsigned packages:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 5E6DA83306132997
```

2. Add the Ice repository to your system:

```
sudo apt-add-repository "deb http://zeroc.com/download/Ice/3.6/ubuntu`lsb_release -rs` stable main"
```

3. Update the package list and install:

```
sudo apt-get update
sudo apt-get install zeroc-ice-all-runtime zeroc-ice-all-dev
```

Refer to the [package summary](#) if you would like to install fewer packages.

The metapackages `zeroc-ice-all-runtime` and `zeroc-ice-all-dev` reference the required Berkeley DB dependencies from the Universe repository. You need to install `db5.3-util` separately to perform admin tasks such as hot backups on your Berkeley DB databases with command-line utilities:

```
sudo apt-get install db5.3-util
```

4. Install the source package (optional):

```
sudo apt-get source zeroc-ice3.6
```

Installing Ice on Debian 9 (stretch)

Ice is included in the Debian 9 (stretch) distribution: <https://packages.debian.org/source/stretch/zeroc-ice>.

You can install all Ice packages with:

```
sudo apt-get install zeroc-ice-all-runtime zeroc-ice-all-dev
```

Refer to the [package summary](#) if you would like to install fewer packages.

The metapackages `zeroc-ice-all-runtime` and `zeroc-ice-all-dev` reference the required Berkeley DB dependencies. You need to install `db5.3-util` separately to perform admin tasks such as hot backups on your Berkeley DB databases with command-line utilities:

```
sudo apt-get install db5.3-util
```



Debian 9 provides Ice packages for many architectures, including amd64 and armhf.

Installing Ice on Red Hat Enterprise Linux 7

Follow the instructions below to install Ice on RHEL 7:

1. Add the Ice repositories to your system:

```
cd /etc/yum.repos.d
sudo wget https://zeroc.com/download/Ice/3.6/el7/zeroc-ice3.6.repo
```

2. Install Ice:

```
sudo yum install ice-all-runtime ice-all-devel
```

Refer to the [package summary](#) if you would like to install fewer packages.

The [yum documentation](#) provides more information about installing packages on RHEL 7.

Notes:

- Although RHEL 7 is a 64-bit operating system, we continue to support 32-bit applications by providing i686 versions of IceBox, IceStorm, C++ run-time libraries, and C++ development libraries, for example:

```
sudo yum install libzeroc-ice-dev.i686
```

- This distribution depends on the Berkeley DB packages included with RHEL 7:

```
sudo yum list 'libdb*'
```

You will need to install `libdb-utils` to perform admin tasks such as hot backups on your Berkeley DB databases with command-line utilities:

```
sudo yum install libdb-utils
```

Installing Ice on Red Hat Enterprise Linux 6

Follow the instructions below to install Ice on RHEL 6:

1. Add the Ice repositories to your system:

```
cd /etc/yum.repos.d
sudo wget https://zeroc.com/download/Ice/3.6/el6/zeroc-ice3.6.repo
```

2. Install Ice:

```
sudo yum install ice-all-runtime ice-all-devel
```

Refer to the [package summary](#) if you would like to install fewer packages.

The [yum documentation](#) provides more information about installing packages on RHEL 6.

Notes:

- This distribution depends on Berkeley DB packages included in our third-party repository:

```
sudo yum list 'db53*'
```

You will need to install `db53-utils` to perform admin tasks such as hot backups on your Berkeley DB databases with command-line utilities:

```
sudo yum install db53-utils
```

Installing Ice on Amazon Linux

Follow the instructions below to install Ice on Amazon Linux:

1. Add the Ice repositories to your system:

```
cd /etc/yum.repos.d
sudo wget https://zeroc.com/download/Ice/3.6/amzn1/zeroc-ice3.6.repo
```

2. Install Ice:

```
sudo yum install ice-all-runtime ice-all-devel
```

Refer to the [package summary](#) if you would like to install fewer packages.

The [EC2 documentation](#) provides more information about installing packages on Amazon Linux.

Notes:

- This distribution depends on Berkeley DB packages included in our third-party repository:

```
sudo yum list 'db53*'
```

You will need to install `db53-utils` to perform admin tasks such as hot backups on your Berkeley DB databases with command-line utilities:

```
sudo yum install db53-utils
```

Installing Ice on SUSE Linux Enterprise Server 12

Follow the instructions below to install Ice on SLES 12:

1. Add the Ice repositories to your system:

```
sudo rpmkeys --import https://zeroc.com/download/GPG-KEY-zeroc-release-5E6DA83306132997
wget https://zeroc.com/download/Ice/3.6/sles12/zeroc-ice3.6.repo
sudo zypper addrepo zeroc-ice3.6.repo
```

2. Install Ice:

```
sudo zypper install ice-all-runtime ice-all-devel
```

Refer to the [package summary](#) if you would like to install fewer packages.

Notes:

- This distribution depends on Berkeley DB packages included in our third-party repository:

```
sudo zypper search 'db53*'
```

You will need to install `db53-utils` to perform admin tasks such as hot backups on your Berkeley DB databases with command-line utilities:

```
sudo zypper install db53-utils
```

Installing Ice on SUSE Linux Enterprise Server 11

Follow the instructions below to install Ice on SLES 11:

1. Add the Ice repositories to your system:

```
wget https://zeroc.com/download/Ice/3.6/sles11/zeroc-ice3.6.repo
sudo zypper addrepo zeroc-ice3.6.repo
```

2. Install Ice:

```
sudo zypper install ice-all-runtime ice-all-devel
```

Refer to the [package summary](#) if you would like to install fewer packages.

Notes:

- This distribution depends on Berkeley DB packages included in our third-party repository:

```
sudo zypper search 'db53*'
```

You will need to install `db53-utils` to perform admin tasks such as hot backups on your Berkeley DB databases with command-line utilities:

```
sudo zypper install db53-utils
```

Setting up your Linux 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++

A C++ program needs to link with at least `libIce` and `libIceUtil`, so a typical link command would look like this:

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

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



Always use the `-pthread` option when compiling and linking your Ice applications.

C++11 Libraries

If you build your C++ program in C++11 or C++14 mode (with `-std=c++0x`, `-std=gnu++0x`, `-std=c++11`, `-std=gnu++11`, `-std=c++14`, `-std=gnu++14`), and this mode is not your compiler's default mode, we recommend you link your application with the Ice C++11 libraries included in your binary distribution. For 32-bit programs:

```
c++ -std=c++11 -o myprogram myprogram.o -pthread -L/usr/lib/c++11 -lIce -lIceUtil
```

For 64-bit programs:

```
c++ -std=c++11 -o myprogram myprogram.o -pthread -L/usr/lib64/c++11 -lIce -lIceUtil
```



When you build your source code with the C++11 or C++14 mode but don't use any of the extra C++ functions that Ice exposes in C++11 mode, you can either:

- link with the Ice C++11 libraries (recommended),
or
- link with the default (C++98) Ice libraries, but only if you also compile the code generated by `slice2cpp` in C++98 mode

Linking with Static Libraries

The Ice C++ packages always include shared libraries. On Debian 8 (jessie), the Ice C++ package also includes static libraries.

The static library `libIce.a` contains the symbols for Ice and IceUtil, and also for all Ice plug-ins (IceSSL, IceDiscovery and IceLocatorDiscovery). Other libraries (`libIceGrid.a`, `libIceStorm.a` etc.) contain symbols for applications that call IceGrid, IceStorm, etc.

Use the following command to link your application with the static Ice library (`libIce.a`):

```
c++ -o client Hello.o Client.o -Wl,-Bstatic -lIce -Wl,-Bdynamic -lcrypto -lssl -pthread -lbz2 -ldl -lrt
```

For an application built with `--std=c++11`:

```
c++ --std=c++11 -o client Hello.o Client.o -Wl,-Bstatic -L/usr/lib/x86_64-linux-gnu/c++11/lib -lIce -Wl,-Bdynamic -lcrypto -lssl -pthread -lbz2 -ldl -lrt
```



When linking with the static Ice library, you need to call [register functions](#) to load Ice plug-ins.

PHP

The Ice extension for PHP is loaded automatically when the interpreter loads the contents of the file `/etc/php.d/ice.ini` (on Red Hat Enterprise Linux and Amazon Linux) or `/etc/php5/conf.d/ice.ini` (on SUSE Linux Enterprise Server and Ubuntu). This file contains the line shown below:

```
extension=IcePHP.so
```

You can modify this file to include additional [configuration directives](#).

At run time, the PHP interpreter requires the Ice shared libraries.

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/share/php` on RHEL, Amazon Linux, and Ubuntu, and in `/usr/share/php5` on SLES). This installation directory is included in PHP's default include path, which you can verify by executing the following command:

```
php -i | grep include_path
```

If the installation directory is listed, no further action is necessary to make the run-time source files available to your application. Otherwise, you can modify the `include_path` setting in `php.ini` to add the installation directory:

```
include_path = /usr/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/share/php');
require 'Ice.php'; // Load the core Ice run time definitions.
```

SELinux Notes (for Red Hat Enterprise Linux users)

SELinux augments the traditional Unix permissions with a number of new features. In particular, SELinux can prevent the `httpd` daemon from opening network connections and reading files without the proper SELinux types.

If you suspect that your PHP application does not work due to SELinux restrictions, we recommend that you first try it with SELinux disabled. As root, run:

```
setenforce 0
```

to disable SELinux until the next reboot of your computer.

If you want to run `httpd` with the Ice extension and SELinux enabled, you must do the following:

1. Allow `httpd` to open network connections:

```
setsebool httpd_can_network_connect=1
```

You can add the `-P` option to make this setting persistent across reboots.

2. Make sure any `.ice` file used by your PHP scripts can be read by `httpd`. The enclosing directory also needs to be accessible. For example:

```
chcon -R -t httpd_sys_content_t /opt/MyApp/slice
```

For more information on SELinux in Red Hat Enterprise Linux, refer to this [Red Hat document](#).

Using the sample programs on Linux

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 Linux

You can launch IceGrid Admin with the `icegridgui` command. IceGrid Admin is a Java program and requires JRE 7u6 or later.

Startup scripts for IceGrid and Glacier2 services

All distributions include the following service configuration files:

- `/etc/icegridregistry.conf`
- `/etc/icegridnode.conf`
- `/etc/glacier2router.conf`

Distributions which use `sysvinit` or `upstart` contain the following sample scripts:

- `/etc/init.d/icegridregistry`
- `/etc/init.d/icegridnode`

- `/etc/init.d/glacier2router`

Distributions which use `systemd` contain the following services:

- `icegridregistry.service`
- `icegridnode.service`
- `glacier2router.service`

The installation also creates a user account and group for running these services (account `ice` and group `ice`), and data directories for `icegridregistry` and `icegridnode` (`/var/lib/ice/icegrid/registry` and `/var/lib/ice/icegrid/node1`).

By default, all these services are off at all runlevels. You need to manually switch on one or more runlevels, for example:

```
#
# On a Red Hat Enterprise Linux 6 system, configure the icegridregistry
# to start at runlevel 3 and 5:
#
sudo chkconfig --level 35 icegridregistry on

#
# On a SUSE Linux Enterprise Server 11 system, configure the
# icegridregistry to start at runlevel 3 and 5:
#
sudo chkconfig icegridregistry 35

#
# On an Ubuntu system, configure the icegridregistry to start at the
# default run levels:
#
sudo update-rc.d icegridregistry defaults

#
# On systems using systemd, start icegridregistry with the multi-user target
#
sudo systemctl enable icegridregistry.service
```

Before doing so, please review the script itself and its associated configuration file.