

# Using the Linux Binary Distributions

This page provides important information for users of the Ice binary distributions on Linux platforms. You can obtain these distributions at the [ZeroC web site](#).

On this page:

- [Overview of the Ice Binary Distributions for Linux](#)
  - [DEB Packages](#)
  - [RPM Packages](#)
  - [Bi-Arch Support on RHEL 7](#)
- [Installing the Linux Distributions](#)
  - [Installing Ice on Ubuntu](#)
  - [Installing Ice on Red Hat Enterprise Linux 7](#)
  - [Installing Ice on Amazon Linux](#)
  - [Installing Ice on SUSE Linux Enterprise Server 12](#)
- [Setting Up your Linux Environment to Use Ice](#)
  - [C++](#)
  - [PHP](#)
- [Using the Sample Programs on Linux](#)
- [Starting IceGrid GUI on Linux](#)
- [Startup Scripts for IceGrid and Glacier2 Services](#)

## Overview of the Ice Binary Distributions for Linux

### DEB Packages

ZeroC provides the following DEB 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-ice-dev</code>	C++ header files and libraries
<code>libzeroc-ice3.7-java</code>	Ice for Java JAR files
<code>libzeroc-ice3.7</code>	C++ run-time libraries
<code>libzeroc-icestorm3.7</code>	IceStorm service for IceBox C++
<code>php-zeroc-ice</code>	PHP extension and run-time files
<code>python3-zeroc-ice</code>	Python extension and run-time files
<code>zeroc-glacier2</code>	Glacier2 service
<code>zeroc-icebox</code>	IceBox server for C++
<code>zeroc-icebridge</code>	IceBridge service
<code>zeroc-ice-compilers</code>	Slice compilers, such as <code>slice2cpp</code> , <code>slice2java</code> and <code>slice2php</code>
<code>zeroc-icegrid</code>	IceGrid service
<code>zeroc-icegridgui</code>	IceGrid GUI application
<code>zeroc-icepatch2</code>	IcePatch2 service
<code>zeroc-ice-slice</code>	Slice files
<code>zeroc-ice-utils</code>	Utilities necessary for administering an Ice installation

[Back to Top](#) ^

### 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
glacier2	Glacier2 service
icebox	IceBox server for C++
icebridge	IceBridge service
icegrid	IceGrid service
icegridgui	IceGrid GUI application
icepatch2	IcePatch2 executable
ice-compilers	Slice compilers, such as <code>slice2cpp</code> , <code>slice2java</code> and <code>slice2py</code>
ice-slice	Slice files
ice-utils	Utilities necessary for administering an Ice installation
libice-c++-devel	C++ header files and symbolic links to the C++ run-time libraries
libice3.7-c++	C++ run-time libraries
libicestorm3.7	IceStorm service
php-ice	PHP extension and run-time files
python-ice	Python extension and run-time files

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

RPM	Description
lmbd	Admin tools for LMDB (statically linked)
lmbd-devel	Header file and static library for LMDB
mcpp-devel	Static library for the MCPP C++ preprocessor



The RPM distribution no longer includes an RPM with the Ice for Java JAR files. See [Building Ice Applications in Java](#) for more information.

[Back to Top ^](#)

## Bi-Arch Support on RHEL 7

On RHEL 7, all of the Ice packages listed above are provided for the `x86_64` architecture, along with a limited subset of packages for the `i686` 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` (for the IceBox services built with the C++98 mapping) and `icebox32++11` (for IceBox services built with the C++11 mapping).

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.

[Back to Top ^](#)

## Installing the Linux Distributions

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 B6391CB2CFBA643D
```

2. Add the Ice repository to your system:

```
sudo apt-add-repository "deb http://zeroc.com/download/Ice/3.7/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.

4. Install the source package (optional):

```
sudo apt-get source zeroc-ice3.7
```

[Back to Top ^](#)

## 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/3.7/el7/zeroc-ice3.7.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.

[Back to Top ^](#)

## 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/3.7/zeroc-ice3.7.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.

[Back to Top ^](#)

# 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:

```
wget https://zeroc.com/download/Ice/3.7/sles12/zeroc-ice3.7.repo
sudo zypper addrepo zeroc-ice3.7.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.

[Back to Top ^](#)

## 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++

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

#### C++11

```
c++ -c -DICE_CPP11_MAPPING -pthread myprogram.cpp
```

#### C++98

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

C++11 and C++98 in the tabs above correspond to the [Ice C++ mapping you're using](#).

When linking a program you must link with at least the Ice library. A typical link command would look like this:

#### C++11

```
c++ -o myprogram myprogram.o -pthread -lIce++11
```

#### C++98

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

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

[Back to Top ^](#)

### 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=ice.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.
```

[Back to Top ^](#)

## 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](#).

[Back to Top ^](#)

## Using the Sample Programs on Linux

Sample programs for all programming languages are available in a separate [ice-demos GitHub repository](#). Simply clone this repository:

```
git clone -b 3.7 https://github.com/zeroc-ice/ice-demos.git  
cd ice-demos
```

[Back to Top ^](#)

# Starting IceGrid GUI on Linux

You can launch IceGrid GUI with the `icegridgui` command. IceGrid GUI is a Java 8-based application.

[Back to Top ^](#)

## 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, as shown below:

```
#
# On systems using svsvinit, configure the icegridregistry to start at the
# default run levels:
#
sudo chkconfig icegridregistry on

#
# 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.

[Back to Top ^](#)