

Building Ice for PHP

This page describes how to build and install Ice for PHP from source code. If you prefer, you can also download [binary distributions](#) for the supported platforms.

On this page:

- [PHP Build Requirements](#)
- [Building the PHP Extension](#)
 - [Building Ice for PHP on Linux and OS X](#)
 - [Building Ice for PHP on Windows](#)
- [Installing the PHP Extension](#)
 - [Installing Ice for PHP on Linux & OS X](#)
 - [Installing Ice for PHP on Windows](#)
- [PHP Dependencies](#)
- [PHP Source Files](#)
- [Running the PHP Tests](#)
- [Web Server Permissions](#)
- [SELinux Notes for PHP](#)

PHP Build Requirements

Ice for PHP is expected to build and run properly on Windows and any recent Linux distribution for x86 and x86_64, and was extensively tested using the operating systems and PHP versions listed on our [platforms page](#).

You will also need the Ice 3.5.1 development kit for C++, which you can install as a binary distribution or compile from source yourself.

Building the PHP Extension

Building Ice for PHP on Linux and OS X

This source distribution only supports building Ice for PHP as a dynamic PHP extension; the product of the build is a shared library that you must configure PHP to load.

First, change to Ice for PHP source subdirectory:

```
$ cd Ice-3.5.1/php
```

Edit `config/Make.rules` and review the build settings. For example, you may want to enable `OPTIMIZE`. If your PHP installation resides in a non-standard location, modify the `PHP_HOME` setting to contain the installation directory. If you are using PHP 5.3 or later and wish to use PHP namespaces, set `USE_NAMESPACES=yes`.

If you have not built Ice for C++ from the `cpp` subdirectory, then set the `ICE_HOME` environment variable to the directory containing your Ice installation. For example, if Ice is installed in `/opt/Ice-3.5.1`, set `ICE_HOME` as follows:

```
$ export ICE_HOME=/opt/Ice-3.5.1
```

If you installed Ice using RPMs, set `ICE_HOME` as shown below:

```
$ export ICE_HOME=/usr
```

Run `make` to build the extension.

Building Ice for PHP on Windows

The Ice installer for Windows includes a precompiled extension for PHP 5.4.8 built with Visual Studio 2008 (Visual C++ 9) and compatible with the "Thread-safe VC9" distribution of PHP available on [php.net](#). We encourage you to use this extension if possible, together with a compatible Web server.



The Apache binaries provided by the Apache Software Foundation are compiled with Visual C++ 6 and therefore are **not** compatible with the Ice for PHP extension included in our Windows installer. A VC9 build of Apache is available from alternate sources.

The most common motivation for compiling the Ice extension yourself is to create an extension that is compatible with your existing environment, such as when your Web server or PHP interpreter is built with a different compiler.

To build the Ice extension, first download the PHP5 binary archive or Windows installer and install the distribution.

You will also need to download and extract the PHP5 sources (Ice for PHP requires the PHP header files). If you are using PHP 5.4, change to the PHP5 source directory and run the following commands:

```
> buildconf
> configure
```

You do not need to actually build PHP from source, but these two steps are necessary to generate header files that are required by the Ice extension.

If you have not built Ice for C++ from the `cpp` subdirectory then you need to set the `ICE_HOME` environment variable to the location of your Ice for C++ installation. For example:

```
> set ICE_HOME=C:\Program Files\ZeroC\Ice-3.5.1
```

Change to the Ice for PHP source directory:

```
> cd Ice-3.5.1\php
```

Open `config\Make.rules.php`, review the comments, and make any necessary changes. In particular, you may need to change the values of `PHP_HOME` and `PHP_BIN_HOME` to refer to your PHP source and binary installations, respectively. If you compiled PHP from source, you should also review the setting of `PHP_LIBDIR`. If you are using PHP 5.3 or later and wish to use PHP namespaces, set `USE_NAMESPACES=yes`. Finally, if you are using a non-thread-safe PHP installation, set `PHP_ZTS=no`.

Run `NMAKE` to build the extension:

```
> nmake /f Makefile.mak
```

Upon successful completion, the Ice for PHP extension is created as `lib\php_ice.dll` (Release) or `lib\php_iced.dll` (Debug).

Installing the PHP Extension

Installing Ice for PHP on Linux & OS X

To install the Ice extension, you must move the extension's shared library into PHP's extension directory. This directory is determined by the PHP configuration directive `extension_dir`. You can determine the default value for this directive by running the command-line version of PHP with the `-i` option:

```
$ php -i
```

Review the output for a line like this:

```
extension_dir => /usr/lib/php/modules => /usr/lib/php/modules
```

Once you've copied the extension to PHP's extension directory, you will need to enable the extension in your PHP configuration. Your PHP installation likely supports the `/etc/php.d` configuration directory, which you can verify by examining the output of `php -i` and looking for the presence of `--with-config-file-scan-dir` in the "Configure Command" entry. If present, you can create a file in `/etc/php.d` that contains the directive to load the Ice extension. For example, create the file `/etc/php.d/ice.ini` containing the following line on Linux:

```
extension = IcePHP.so
```

Or on OS X:

```
extension = IcePHP.dylib
```

If PHP does not support the `/etc/php.d` directory, determine the path name of PHP's configuration file as reported by the `php -i` command:

```
Configuration File (php.ini) Path => /etc/php.ini
```

Open the configuration file and append the following line on Linux:

```
extension = IcePHP.so
```

Or on OS X:

```
extension = IcePHP.dylib
```

You can verify that PHP is loading the Ice extension by running the command shown below:

```
$ php -m
```

Look for `ice` among the installed modules. Note that your library search path (`LD_LIBRARY_PATH` on Linux or `DYLD_LIBRARY_PATH` on OS X) must include the directory containing the Ice shared libraries.

Read the [PHP Dependencies](#) and [PHP Source Files](#) sections below for more information about installing the Ice extension.

Installing Ice for PHP on Windows

To install the Ice extension, you must move the extension's shared library into PHP's extension directory. This directory is determined by the PHP configuration directive `extension_dir`. You can determine the default value for this directive by running the command-line version of PHP with the `-i` option:

```
> php -i
```

Review the output for a line like this:

```
extension_dir => C:\php5 => C:\php5
```

The directive may also be set to a relative path, such as:

```
extension_dir => ./ => ./
```

In the case of a relative path, the value is relative to the current working directory of the process. As a result, the working directory when running the command-line version of PHP will likely differ from the working directory when PHP is running as a Web server module. Using a typical installation of Apache as an example, the working directory is Apache's installation directory, therefore the extension must be copied to

```
\Program Files\Apache Software Foundation\Apache2.2
```

Once you've copied the extension to the appropriate directory, you will need to enable the extension in your PHP configuration. First you must discover the location of PHP's configuration file (`php.ini`), which is also displayed by the `-i` option. Look for the following line:

```
Loaded Configuration File => C:\Program Files\PHP\php.ini
```

If you used the Windows installer for PHP, your Web server's configuration may have already been modified to load PHP. You can also review your Web server's settings to discover the location of `php.ini`. For example, PHP's Windows installer modifies Apache's configuration to add the following directives:

```
PHPIniDir "C:/Program Files/PHP/"
LoadModule php5_module "C:/Program Files/PHP/php5apache2_2.dll"
```

The `PHPIniDir` directive specifies the directory containing the `php.ini` file.

Open `php.ini` and append this directive:

```
extension = php_ice.dll
```

Read the [PHP Dependencies](#) and [PHP Source Files](#) sections below for more information about installing the Ice extension.

PHP Dependencies

PHP will need to be able to locate the libraries for the Ice run-time libraries and its third-party dependencies. On Unix-style platforms, these libraries are named as follows:

```
libIce
libIceUtil
libSlice
libbz2
```

On Windows, these DLLs are required:

```
icevc90_35.dll
iceutilvc90_35.dll
slicevc90_35.dll
bzip2_vc90.dll
```

In general, these libraries must reside in a directory of the user's `PATH`. For Web servers, the libraries may need to reside in a system directory. For example, on Linux you can add the directory containing the Ice run-time libraries to `/etc/ld.so.conf` and run `ldconfig`. On Windows, you can copy the DLLs to the `C:\WINDOWS\system32` directory, or to the Apache installation directory.

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. For example, you can create a file in the Web server's document directory containing the following PHP script:

```
<?php
phpInfo();
?>
```

Then start a browser window and open the URL corresponding to this script. If the Ice extension is successfully installed, you will see an `ice` section among the configuration information.

Note that if you want to use IceSSL from the Ice extension, then PHP will also need access to the shared libraries for IceSSL and OpenSSL.

PHP Source Files

In addition to the binary Ice extension module and its library dependencies, you will also need to make the Ice for PHP source files available to your scripts. These files are located in the `lib` subdirectory and consist of the Ice run time definitions (`Ice.php` or `Ice_ns.php`) along with PHP source files generated from the Slice files included in the Ice distribution.

The Ice extension makes no assumptions about the location of these files, so you can install them anywhere you like. For example, you can simply include them in the same directory as your application scripts. Alternatively, if you prefer to install them in a common directory, you may need to modify PHP's `include_path` directive so that the PHP interpreter is able to locate these files. Another option is to modify the include path from within your script prior to including any Ice run-time file. Here is an example that assumes Ice is installed in `/opt`:

```
// PHP
ini_set('include_path',
ini_get('include_path') . PATH_SEPARATOR . '/opt/Ice-3.5.1/php');
require 'Ice.php'; // Load the core Ice run time definitions.
```

Running the PHP Tests

The `test` subdirectory contains PHP implementations of the core Ice test suite. [Python](#) is required to run the test suite.

The test suites require that the Ice for C++ tests be built in the `cpp` subdirectory of this source distribution. In addition, the scripts require that the CLI version of the PHP interpreter be available in your `PATH`.

To run all of the tests, do the following:

```
> python allTests.py
```

You can also run tests individually by changing to the test directory and running this command:

```
> python run.py
```

If everything worked out, you should see lots of "ok" messages. In case of a failure, the tests abort with "failed".

Web Server Permissions

The Web server normally runs in a special user account that may not necessarily have access to the Ice extension, its dependent libraries and PHP source files, and other resources such as Ice configuration and your application scripts. It is very important that you review the permissions of these files and verify that the Web server has sufficient access.

For example, on Windows the Apache server typically runs as a service in the "Local System" account. You will need to modify the access rights of the aforementioned files to grant access to this account. In a command window, you can use the `cacls` utility to establish the appropriate access rights. Assuming that you have copied the Ice extension and dependent DLLs to Apache's installation directory, you can modify the access rights as shown below:

```
cd \Program Files\Apache Software Foundation\Apache2.2
cacls php_ice.dll /G SYSTEM:F Administrators:F
cacls bzip2.dll /G SYSTEM:F Administrators:F
...
```

On Linux, Apache typically runs in the `apache` account, so you will either need to change the owner or group of the libraries and other resources, or modify their permissions to make them sufficiently accessible.

SELinux Notes for PHP

[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 Ice for 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 Ice for PHP and SELinux enabled, there are two steps you need to take. First, **allow httpd to open network connections**:

```
# setsebool httpd_can_network_connect=1
```

Add the `-P` option to make this setting persistent across reboots.

Second, 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
```