

# Known Problems in Ice 3.5

This page describes the known problems Ice 3.5.

On this page:

- [Parameter limit for C++11 lambda functions with Visual C++](#)
- [Drag and drop failures in IceGrid Admin GUI](#)
- [Socket connection issue in Android emulator](#)
- [SSL issues in Java](#)
- [SSL issues in Android](#)
- [Visual Studio Add-in issues with Visual Studio 2010](#)
- [Solaris Java 7 issue](#)
- [SSL certificate issues with .NET on Windows XP](#)
- [IPv6 on Windows XP](#)
- [IPv6 hang in Java](#)

## Parameter limit for C++11 lambda functions with Visual C++

Ice 3.5 supports the use of [lambda functions as AMI callbacks](#) for C++11 compilers, including Visual Studio 2010 and 2012. However, neither compiler supports [variadic templates](#), and the macros the compilers use to simulate this feature are limited to at most ten parameters. (Visual Studio 2012 actually has a default limit of five parameters, but this can be increased to as many as ten by defining `_VARIADIC_MAX`.) As a result, you can only use the type-safe callback API for Slice operations containing ten or fewer output parameters, including a non-void return value; operations that exceed this limit cause a compilation error. The generic callback API is not affected by this issue. Ice 3.5.1 includes a work-around that disables the type-safe callback API for operations with more than ten output parameters.



Visual Studio 2013 includes true support for variadic templates. There is no parameter limit for C++11 lambda functions when using this compiler.

## Drag and drop failures in IceGrid Admin GUI

A JavaFX bug can generate `NullPointerException` errors when using drag & drop in the [IceGrid graphical administration tool](#). This bug is fixed in Java7u10.

## Socket connection issue in Android emulator

An Ice application that attempts to connect to an invalid address can generate "spurious wakeup" messages in logcat when running under an emulator for Android 4.0 or later. This issue does not occur on hardware devices, however you can still experience lengthy delays before receiving an exception. As a defensive measure, it is a good idea to always set reasonable timeouts on your proxies to avoid unexpected delays.

## SSL issues in Java

When using the Ice for Java SSL plug-in (IceSSL), you may experience occasional hangs. The most likely reason is that your system's entropy pool is empty. If you have sufficient system privileges, you can solve this issue by editing the file `java.home/jre/lib/security/java.security` and changing it to use `/dev/urandom` instead of `/dev/random`. If you do not have permission to modify the security file, you can also use the comma and-line option shown below:

```
java -Djava.security.egd=file:/dev/urandom MyClass ...
```

## SSL issues in Android

Ice for Android supports the use of SSL on Android 2.3 or later.

A limitation in Android 2.3 prevents the use of client certificates. Ice servers that support Android 2.3 clients must set `IceSSL.VerifyPeer=0` to allow an incoming connection to succeed without requiring the client to supply a certificate. The [property reference](#) discusses this setting in more detail.

## Visual Studio Add-in issues with Visual Studio 2010

If you perform a custom Visual Studio 2010 installation selecting only "Visual C++" development tools, the Visual Studio Add-in will fail to load with an exception trying to load stdole assembly. The recommended workaround is to add "Visual C#" development tools in the Visual Studio 2010 installer.

## Solaris Java 7 issue

On Solaris with Java 7, an Ice application might print "spurious selector wake up" warnings. This is caused by the Java selector API not behaving as expected and can impact the performance of your Ice application. If you experience performance issues but do not see these warnings, it is also possible that you are affected by this issue as the warning only shows up after multiple spurious wake ups. Upgrading to Java 7u10 fixes this problem.

## SSL certificate issues with .NET on Windows XP

The SSL certificates included in the Ice source and demo distributions use the SHA256 hash algorithm for compatibility with iOS. Unfortunately, this algorithm is not supported when using .NET on Windows XP, so you will not be able to use the SSL-related .NET demos on this platform.

## IPv6 on Windows XP

Ice enables IPv6 by default on all platforms. On Windows XP, where IPv6 might not be enabled or supported (Java does not support IPv6 on Windows XP), this will cause an `Ice::SocketException` when Ice tries to create an IPv6 socket. To workaround this issue, you should explicitly disable IPv6 by setting the [Ice.Ipv6](#) property to 0.

## IPv6 hang in Java

On systems with IPv6 enabled, you may experience [occasional hangs](#) the first time an Ice object adapter is activated within a JVM. A work-around is to disable IPv6 support by setting the Java property `java.net.preferIPv4Stack` to true. For example:

```
java -Djava.net.preferIPv4Stack=true MyClass ...
```