Known Problems in Ice 3.4.2

This page describes the known problems Ice 3.4.2.

On this page:

`iceca` failure on SLES 11

The Ice Certificate Authority (`iceca`) script may fail to import OpenSSL-generated certificates into a Java keystore. This failure occurs when using the following command-line options:

```
iceca java --import ...
```

The import fails with an error similar to:

```
lengthTag=127, too big
```

The error is caused by an incompatibility between the JDK's keytool and the version of OpenSSL that is included with SLES11 (OpenSSL 0.9.8h). To work around this issue, you can install a newer version of OpenSSL. Note that it is not necessary to rebuild Ice with the new OpenSSL version; the only requirement is that you have the new `openssl` executable in your `PATH` when running the command `iceca java --import ...`.

Socket issue in Android 2.2

An Ice application that works correctly in Android 2.1 may fail in Android 2.2 with a socket exception:

```
java.net.SocketException: Bad address family
```

This exception is caused by Android bug 9431. To work around it, add the following code to your application's initialization logic:

Java

```
if(android.os.Build.VERSION.SDK_INT == 8) // FROYO (8)
{
    java.lang.System.setProperty("java.net.preferIPv4Stack", "true");
    java.lang.System.setProperty("java.net.preferIPv6Addresses", "false");
}
```

The `if` statement ensures that the code is only executed when running on Android 2.2.

SSL issues in Android

Ice for Android supports the use of SSL on Android 2.2 or later. In the sample programs, SSL is disabled when using Android 2.1 or earlier due to bug 4914.

Note that there is an SSL incompatibility between an Android client and an Ice for C++ (OpenSSL) server that prevents a connection from being established in certain situations. If the server is configured to request and validate the client's certificate, which occurs when the server's `IceSSL.VerifyPeer` property is set to 1 or 2, Android raises an exception indicating a "failed handshake". The only known solution at this time is to set `IceSSL.VerifyPeer=0` in the server, which allows the connection to succeed but has the disadvantage that the client's certificate is no longer validated.

Also note that SSL incurs significantly more overhead than TCP in Android. Connection establishment in particular is very costly.