

# libjingle Developer Guide

The libjingle SDK consists of C++ source code and documentation that enable you to design applications that connect and exchange data across a network (peer to peer data connections). The SDK contains code and sample applications, a Visual Studio 2005 solution file for compiling on Windows computers, configuration and makefiles for other operating systems and compilers, and compilation instructions. Note that the code has some external dependencies (for instance, the voice chat sample has dependencies on Linphone or GIPS VoiceEngine Lite, depending on your platform). Dependencies are indicated in the code and the build instructions.

The code includes network and proxy negotiation classes, XML parsing classes, a STUN server, and all the code required to initiate connections and exchange data between two computers. The connection code enables applications to robustly traverse NAT and firewall devices using the ICE mechanism, to use STUN servers, and to exchange either UDP or TCP data packets. You can use the code as provided, or extend it fit your own specific needs, according to the Berkeley-style license.

## Jingle and libjingle

libjingle was created at about the same time as the Jingle XMPP extension (XEP-0166). The libjingle team created their own protocol to handle session negotiation, and later worked with the XMPP Standards Foundation to standardize Jingle; thus, although the libjingle protocol and Jingle are very similar, they are not the same, and are not interoperable.

The current version of the libjingle code still uses the original internal protocol, which is slightly different from, and incompatible with, the Jingle specification. Nevertheless, it is close enough to Jingle that it is worthwhile learning the Jingle specifications. Similar "close but not identical" conditions exist for libjingle's audio content description (early version of Jingle Audio Content Description Format XEP-0167), ICE transport description (early version of Jingle Ice Transport XEP-0176), and raw UDP transport description (early version of Jingle Raw UDP transport description XEP-0177). Where this documentation refers to "Jingle" or one of its related extensions (in terms of how libjingle uses that protocol), it really refers to the original, internal protocol

The libjingle team intends to implement the official Jingle extension in libjingle (while making libjingle backward compatible for the current custom version).

## Prerequisites

In order to use libjingle, you should be familiar with the XMPP protocol, general networking concepts, and C++. Additionally, it helps to be familiar with the Jingle proposed extension (XEP-0166), and other related extensions listed in Jingle and libjingle above.

The requirements to build and use libjingle are given in Creating a libjingle Application.

## Organization of the Documentation

This documentation includes the following main topics:

- Important Concepts Describes some of the key concepts in libjingle. Very important if this is your first experience with libjingle.

- Generic libjingle Applications Provides an overview of the main components of a libjingle application, and some specifics about how a generic libjingle application creates a peer-to-peer connection.
- Creating a libjingle Application Describes the key tasks that every libjingle application must perform.
- Sample Applications Describes how to build and run the sample applications included with the SDK. Also includes detailed information about how these applications work.
- Reference A reference guide to the important libjingle classes.

## Organization of the SDK

The libjingle SDK contains the following folders.

| Directory | Description |
|---|---|
| base/ | Classes for low-level base functionality such as sockets and threads. Both the p2p and session components depend on these classes. |
| examples/ | Contains two sample applications built on libjingle: call, and file share. There are other files in this directory, but they are used for helper applications. |
| p2p/ | Classes in libjingle's Peer to Peer component, which negotiate, establish, and maintain peer-to-peer connections through almost any network configuration regardless of NAT-enabled devices and firewalls. |
| session/ | Classes that specializes the behavior of the basic peer to peer session according to what type of data is being exchanged (for example, voice or files). |
| third_party/ | The default folder to hold various third-party extensions needed by the code. For example, Linphone and GIPS VoiceEngine Lite files should be installed here to use in the voice chat sample application. |
| xmllite/ | Classes for parsing and building XMPP stanzas. |
| xmpp/ | Classes for sending and receiving XMPP requests, and managing common XMPP server tasks (such as logging in or announcing presence). |

**Notes** libjingle has some issues you should be aware of:

- libjingle code is a work in progress. It is not perfect, complete, or flawless. Some aspects of the code, such as password protection, are not securely implemented, and others, such as URL parsing, are not as robust as they should be. Many, but not all, of these issues are commented in the code. Be sure to code your application as robustly as possible. See Terms and Conditions for the legal mumbo jumbo.
- The example code uses SSL to transmit the password from the client to the XMPP server; however, it does not provide any additional security (except turning off screen echo) to protect the password on the client.



*Last updated March 23, 2012.*