Sample Applications

Libjingle ships with two command-line sample applications:

- **call** This is a voice chat sample application.
- **pcp** This is a file-sharing sample application.

The details of how these sample applications work are given in the linked sections. Both samples use a helper solution, **login**. This a helper application used by both call and the file share sample application to sign into a server. It is not expected to be run alone.

Installation and compilation directions are provided in the readme (UNIX/Linux) and readme.win (for Windows Visual C++ 2005 Express Edition). The Windows build currently requires that you install the Platform SDK.

> **Warning** The sample applications do not provide any security to protect the user's password on the computer. The password is protected by SSL during transmission to the server, but is not protected in memory on the computer itself (other than by not echoing the password as it is typed in). If password security is crucial to your implementation, consider providing additional security to the password handling code.

Build flags

The following #define flag values are important to know when building your application:

- HAVE_GIPSLITE Defined if you have GIPS VoiceEngine Lite installed on your computer.
- NO_ATL (Windows only) Defined if your Platform SDK installation does not include ATL. More information is given in the installation and compilation files.
- FEATURE_ENABLE_SSL Define this value if you want to enable SSL encryption for your communications. If you want to intercept and read the packets easily during development, it might be useful to #undefine this value (in Windows this is defined in the Visual Studio build settings, not in the code). The provided code sets defines this value by default. See SSL Support.

The location of these values can be in either the code, the makefile, or the Visual Studio environment, depending on your build environment.

Logging

The sample applications include logging code that print messages to the stdout stream. These messages include incoming and outgoing stanza values and error values. See Logging for more information about changing the logging settings.

Voice Chat Sample Application

The voice chat sample application (call) is a voice chat application based on either GIPS VoiceEngine Lite or Linphone, depending on your operating system. It is a command-line application that provides basic alerts and presence notification. See About Voice Chat Applications for a detailed description about how the call sample application works.

The call sample application has dependencies on Linphone (UNIX/Linux) and GIPS VoiceEngine Lite (Windows). Linphone is included with the installation, but you must download GIPS Voice Engine Lite from the Web site indicated in the Toolbox section of the home page.

The application runs interactively and will prompt you for the required information. You can modify the code to call a different XMPP server; to do this, modify the host and server addresses in the main function in call_main.cc, and change the STUN port server address specified in CallClient::InitPhone. You can talk to another user who is running either call or Google Talk, but you cannot run two instances of the application and call yourself, as you can with the File Share sample application.

File Share Sample Application

The file share sample application (pcp) is a command-line application that sends files from one computer to another. To run this program, you must have a Gmail account, or else you must modify the code to use a different server. For testing purposes, you can send a file to yourself by running two instances of the sample program.

> **Note** If running on the same computer, you should run the executable from different directories (though you can reference the same executable from different locations).

1. Start the file receiver

Start the instance of the program that will receive the files. The client takes no parameters, only options. Specify the -d option to display logging information, which includes XMPP stanzas sent and received, which can be very useful for understanding the program.

> **Note** Received files will be stored in the folder from which you run the program. Make sure that the directory does not include files or folders with the same name as the files or folders, because the program will fail rather than overwrite identically named objects.

```
pcp -d
```

You will receive prompts for your JID and password. Use your bare JID (*username*@*domain.com*). The client will continue running until all files have been transferred, or an error occurs in transfer.

2. Start the file sender with a list of files

Start another instance of the application with the following syntax:

```
pcp [-fd] <space-delimited_file_names> <bare_JID_of_client>
```

The list of files is delimited by spaces. Copying files from parent directories may not work well on Windows systems; for best performance, copy files from the current directory or lower. The client JID should be a bare JID (that is, not qualified by a resource). Use the -d flag to show the stanzas and other logging information processed by the program.

After the files have transferred (or an error has occurred), both client and server will sign off the server and terminate.

For more information about the structure of this program, see the file share application description.

Last updated March 23, 2012.