pc-bluetooth-server Resources

Hard-coded file transfer from PC to target device over Bluetooth connection

- Connect to target device
- Transfer file

We need a working, Windows-based implementation of the Object Exchange (OBEX) protocol for our language of choice for the desktop application.

The OBEX protocol is widely used for network communication between Bluetooth and other devices for tasks such as file transfer.

OFFICIALLY USING JAVA FOR IMPLEMENTATION: BlueCove

Python-implementation-failed

PYBLUEZ

Python extension module allowing access to system Bluetooth resources

- detects, browses, establishes connections
- cannot transfer files

Python implementations of the OBEX protocol

LIGHTBLUE

- not implemented for Windows

PYOBEX

- not implemented for Windows

Java-implementation-in-progress

JSR-82

Java APIs for Bluetooth Wireless Technology.

Sun Java Wireless Toolkit

A set of tools for creating Java applications that run on devices compliant with the Java Technology for the Wireless Industry (JTWI, JSR 185) specification and the Mobile Service Architecture (MSA, JSR 248) specification. It consists of build tools, utilities, and a device emulator.
https://docs.oracle.com/javame/dev-tools/wtk-cldc-2.5.2-01/UserGuide-html/bluetooth.html

Interesting Java Bluetooth Community Guide:
https://community.oracle.com/docs/DOC-983347

Interesting Java code example to check for specific services using UUID:
http://www.aviyehuda.com/blog/2010/01/08/connecting-to-bluetooth-devices-with-java/

Obex
Example OBEX client/server using Bluecove
http://www.bluecove.org/bluecove/apidocs/overview-summary.html#OBEXPutClient

Oracle Obex guide
http://www.oracle.com/technetwork/articles/javame/bluetoothobex-156467.html

Important Distinctions on Input/Output object types in Java

- java.io.Reader: abstract class for reading CHARACTER streams (i.e., InputStreamReader, FileReader, BufferedReader)
- java.io.BufferedReader: concrete class that reads character input from a stream in BUFFERS, which is efficient for characters/arrays/lines
- java.io.InputStream: abstract class for reading an input stream of bytes (ANY data such as images, and non-text based)
- java.io.FileInputStream: concrete class of java.io.InputStream that reads raw bytes from files
- java.io.Writer: abstract class for writing to CHARACTER streams
- java.io.PrintWriter: concrete class for formatting an object as a String to a text-output stream

Decisions:

- For reading input from the Android server, we use BufferedReader since we only accept characters from it.
- For sending data to the Android Server, we use the abstract OutputStream to send all types of bytes (character and raw file bytes)

File Serialization In Java

- Can serialize arrays
- Ideally, we would represent our data in XML or JSON and then serialize
- https://www.wikihow.com/Serialize-an-Object-in-Java
- Use ByteArray(Input/Output)Stream?
- 5 interesting facts of serialization