

Avetana JSR82 Implementation Project description

Development Status: pre-beta

Platforms: Linux + BlueZ, Mac OS X, Windows.

Features:

Bluetooth URLs:

- The class JSR82URL manages all kind of bluetooth URLs and prevents from non-conforming URLs.

RFCOMM

- Non secured RFCOMM connections are available. The user can connect to devices with the help of the javax.microedition.io.Connector class. A typical connection URL is:
btspp://010203040506:1;master=false
The options „encrypt“, „authenticate“, „master“, „authorize“ are not supported
- Non secured RFCOMM servers waiting for remote connections are available. The user can create a new service with the javax.microedition.io.Connector class. A typical connection URL is :

btspp://localhost:3B9FA89520078C303355AAA694238F07:1;name=Avetana Service;

The options „encrypt“, „authenticate“, „master“, „authorize“ are not supported.

The service will be registered in the local bluetooth service Database after the call of the method acceptAndOpen();

The service can be updated with the call of the method „updateRecord“ of the class „LocalDevice“. This method is partially implemented (some verifications are missing).

- The settings of the local device can be retrieved with the help of the class „LocalDevice“:
 - Device Class
 - Friendly name
 - Device properties (see the javadoc of the method getProperty)
 - Discovery mode (the discovery mode can also be set).
 - Bluetooth address

Some of these settings require root privileges (for example the discovery mode).

DiscoveryAgent:

- Device Inquiry: this feature is fully implemented (use of RemoteDevice objects).
- Service Discovery: this feature is implemented but a request can not be cancelled. (the method cancelServiceSearch is not implemented).
All SDP constants are available. 16,32 and 128-bits UUIDs are available. Local and remote Service Record objects can be created with appropriated classes and methods. All verifications requested by the specification are also made.

L2CAP Connections

- The methods getReceiveMTU() and getTransmitMTU() are yet implemented and an L2CAP connection is possible. However, this feature was only quickly tested.

Security

- Authentication, Encryption, role switching (Master/Slave) of an existing connection are implemented under Linux. All these methods requests the user to have root privileges.