The JPen JRE Extension and JPen Bridge

JPen supports another method of deployment besides a signed applet/JNLP. This method was designed to not require signing, which empirically turns users away. It works by inserting a layer of indirection between client programs and JPen called the bridge. The bridge simulates a pen if JPen is missing from the environment and hooks in the correct version of JPen if present.

Every client program that uses the bridge can be in one of two states:

1. JPen is not installed. Pressure, tilt, z, and pen modes will be static, set to default values
2. JPen is installed. All pen values supported by the digitizer will be dynamic

To be in the second state, a user must at some point install JPen as a JRE extension. This can be done by downloading and running an installer program provided in the JPen distribution.

The key to a successful bridged application is being able to operate in the first state well enough for the user to trust the application and see the value in installing the JRE extension. Also, once the JRE extension is installed it never needs to be re-installed.

Versioning Notes

The JRE extension and bridge are paired by SVN revision. That is, a bridge built at revision X will only work with a JRE extension built at version X.

A complete archive of bridges and extension installers can be found at Y.

Code Examples using the JPen Bridge

The key classes that make up the bridge are PenBridge, BridgedPenListener, and BridgedPenEvent.

A simple example:

PenBridge bridge = new PenBridge();

bridge.setDefaultPressure(0.9f);

Bridge.addBridgedPenListener(frame, listener);

// At this point listener is reporting pen or simulated pen events from frame