

Checking network strength in Java ME

This code snippet demonstrates how to retrieve the current network strength.

Overview

The code example retrieves the system property `com.nokia.mid.networkavailability`.

If a network is available, the property will contain the string `available`. After that, the application reads the network strength from the `com.nokia.mid.networksignal` property and displays it.

Note that in Series 40 platform only signed MIDlets under operator domain can retrieve this information.

Source file: NetworkStrength.java

```
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;
import javax.microedition.midlet.MIDlet;

public class NetworkStrength extends MIDlet implements CommandListener{

    private Display display;
    private Form form;
    // Commands.
    private Command exitCommand;
    private Command refreshCommand;

    // Output fields.
    private StringItem strength;
    private StringItem availability;

    /**
     * Constructor. Constructs the object and initializes displayables.
     */
    public NetworkStrength() {
        form = new Form("Network strength.");

        exitCommand = new Command("Exit", Command.EXIT, 2);
        refreshCommand = new Command("Refresh", Command.OK, 1);

        availability = new StringItem("Network status:", "");
        strength = new StringItem("Network signal strength:", "\n --- ");

        form.append(availability);
        form.append(strength);

        form.addCommand(exitCommand);
        form.addCommand(refreshCommand);
        form.setCommandListener(this);
    }
}
```

```
display = Display.getDisplay(this);
display.setCurrent(form);

    checkNetworkStatus();
}

/***
 * From MIDlet.
 * Called when the MIDlet is started.
 */
public void startApp() {
    // No implementation required.
}

/***
 * From MIDlet.
 * Called to signal the MIDlet to enter the Paused state.
 */
public void pauseApp() {
    // No implementation required.
}

/***
 * From MIDlet.
 * Called to signal the MIDlet to terminate.
 * @param unconditional whether the MIDlet has to be unconditionally
 * terminated
 */
public void destroyApp(boolean unconditional) {
    // No implementation required
}

/***
 * From CommandListener.
 * Called by the system to indicate that a command has been invoked on a
 * particular displayable.
 * @param cmd the command that was invoked
 * @param displayable the displayable where the command was invoked
 */
public void commandAction(Command c, Displayable d) {
    if (c == refreshCommand) {
        checkNetworkStatus();
    } else if (c == exitCommand) {
        notifyDestroyed();
    }
}

/***
 * Called to check network availability and strength.
 */
private void checkNetworkStatus() {
    String available =
        System.getProperty("com.nokia.mid.networkavailability");
    String signal = System.getProperty("com.nokia.mid.networksignal");
    //If network available.
    if (available!=null && available.compareTo("available") == 0) {
        availability.setText("\n"+available+"\n");
    }
}
```

```
        strength.setText(signal);
    } else {
        availability.setText("\n unavailable \n");
        strength.setLabel("Error:\n");
        strength.setText("Unable to retrieve current battery level.\n" +
            "It's only available for signed midlets " +
            "under operator domain.");
    }
}
```

Postconditions

If a network is available, the MIDlet displays the current signal strength.

Supplementary material

The source file and executable application are available for download at [Media:Checking network strength in J2ME.zip](#).