Making a phone call using TextField in Java ME

This snippet demonstrates how to make a phone call by using a `TextField`. The MIDlet constructs a phone number field and displays it on the screen. The user may enter a phone number into it. After that, he or she can select `Options` > `Call` to call that number.

Source

```
import javax.microedition.io.ConnectionNotFoundException;
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.TextField;
import javax.microedition.midlet.MIDlet;
```

```
public class MakingPhoneCallTF extends MIDlet implements CommandListener {
    private TextField numberField;
    private Command callCommand;
    private Command exitCommand;
    private Form mainForm;

    /**
     * Constructor. Constructs the object and initializes displayables.
     */
    public MakingPhoneCallTF() {
        mainForm = new Form("Making Phone Call");

        // Create a phone number field, which allows a call to be made
        numberField = new TextField("Phone number", null, 20, TextField.PHONENUMBER);
        mainForm.append(numberField);

        callCommand = new Command("Call", Command.SCREEN, 0);
        mainForm.addCommand(callCommand);

        exitCommand = new Command("Exit", Command.EXIT, 0);
        mainForm.addCommand(exitCommand);
        mainForm.setCommandListener(this);
    }

    /**
     * Called when the MIDlet is started.
     */
    public void startApp() {
        Display.getDisplay(this).setCurrent(mainForm);
    }

    /**
     * Called when MIDlet is paused.
     */
    public void pauseApp() {
    }

    /**
     * Called to signal the MIDlet to terminate.
     *
     * @param unconditional if true, then the MIDlet has to be unconditionally
     * terminated and all resources has to be released.
     */
    public void destroyApp(boolean unconditional) {
    }

    /**
     * From CommandListener.
     * Called by the system to indicate that a command has been invoked on a
     * particular displayable.
     * @param command the command that was invoked
     * @param displayable the displayable where the command was invoked
     */
    public void commandAction(Command command, Displayable displayable) {
        if (command == callCommand) {
            call();
        } else if (command == exitCommand) {
            // Exit the MIDlet
            destroyApp(true);
            notifyDestroyed();
        }
    }

    /**
     * Makes a call to number user typed in the text field.
     */
    private void call() {
        String number;

        number = numberField.getString();
        try {
            platformRequest("tel:" + number);
        } catch (ConnectionNotFoundException ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

Postconditions

A phone number field is displayed on the screen. The user can enter a phone number and select `Options` > `Call` to call that number.

(C) Copyright Nokia 2013. All rights reserved.

See also

- Making a phone call programmatically using platformRequest() in Java ME

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.