Recieving Cell Broadcasting Services (CBS) Messages on Java ME

C.B.S strands for Cell Broadcasting Services. Using J2ME we are able to receive C.B.S message but it is not possible to send C.B.S message. We can receive the C.B.S message which is being broadcast on particular channel. For the service provider like AIRTEL and Vodafone we can receive it on the channel 50. C.B.S message is also part of WMA API (Wireless Messaging API) in Java ME and works in same manner as SMS (Short Message Service) and MMS (Multimedia Messages). The significant difference is in SMS we are using port number to receive messages and for C.B.S we need to register for particular channel Number.

This feature is not supported by all Nokia devices. See here for details.

Process of Receiving C.B.S

1) Device Settings

It is advisable to make settings in device so device can also receive C.B.S. Perform the following steps.

- Go to Message menu
- Go to option and select C.B.S
- Select subscribe and give channel number(50 for the service provider like Airtel and Hutch)

The above are the settings which i have done in E50 if you have other mobile then there might be some change.

2) Register for a particular channel

The first step we need to do is register for particular channel to receive the C.B.S from application.

```
this.connection = (MessageConnection) Connector.open("cbs://:50", Connector.READ);

this.connection.setMessageListener(this);
```

Use the following method of push registration to start the application at the time of receiving C.B.S.

```
Syntax:   registerConnection(String connection, String midlet, String filter)

Example:  pushRegistery.registerConnection("cbs://:50",this,*);
```

Source Code:

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;
import javax.microedition.lcdui.Item;
import javax.wireless.messaging.MessageListener;
import javax.wireless.messaging.MessageConnection;
import javax.wireless.messaging.Message;
import javax.wireless.messaging.TextMessage;
import java.io.IOException;

public final class CbsTestMIDlet extends MIDlet implements MessageListener, Runnable  {
```


(C) Copyright Nokia 2014. All rights reserved.

```
    private MessageConnection connection;
    private Form form;
    private int count;

    protected void startApp() throws MIDletStateChangeException {
        Display display = Display.getDisplay(this);
        form = new Form("CBS Test");
        display.setCurrent(form);
        form.append(appendLine("Open channel..."));
        try {
            this.connection = (MessageConnection) Connector.open("cbs://:50",
Connector.READ);
            this.connection.setMessageListener(this);
            form.append(appendLine("Waiting..."));
        } catch (Throwable t) {
            form.append(appendLine(t.toString()));
            display.setCurrent(new Alert("Error", t.toString(), null, AlertType.ERROR),
form);
        }
    }

    protected void pauseApp() {
    }

    protected void destroyApp(boolean b) throws MIDletStateChangeException {
        // TODO quit politely
    }

    public void notifyIncomingMessage(MessageConnection messageConnection) {
        if (++count > 24) { // 24 msgs max - stop listening...
            try {
                this.connection.setMessageListener(null);
            } catch (IOException e) {
                // ignore
            }
        } else {
            (new Thread(this)).start();
        }
    }

    public void run() {
        try {
            Message message = connection.receive();
            if (message instanceof TextMessage) {
                form.append(appendLine("[" + ((TextMessage) message).getPayloadText() +
"]"));
            } else {
                form.append(appendLine("[<binary msg>]"));
            }
        } catch (IOException e) {
            form.append(appendLine("[" + e.toString() + "]"));
        }
    }

    private Item appendLine(String line) {
        StringItem item = new StringItem(null, line);
```


(C) Copyright Nokia 2014. All rights reserved.

```
        item.setLayout(Item.LAYOUT_2 | Item.LAYOUT_NEWLINE_AFTER);
        return item;
    }
}
```


(C) Copyright Nokia 2014. All rights reserved.