Determining Current Location via Cell ID

This article shows how to get the current location using Cell ID in Nokia Asha and S40 devices.

22 Dec 2013

Introduction

Determining the current location of a device without an integrated GPS or Bluetooth GPS receiver is possible by retrieving the cell ID of the operator's cell which the phone is associated to. The accuracy however of using this method, compared to a standard GPS receiver, is much lower and varies from 100-500 meters for urban areas, down to 10 km for rural areas. The cell ID is the unique identifier of a base station within the same mobile phone network. Using a different operator from the same location means that the device is associated to a different cell. Association between the cell ID and the current location is made by using the API, which makes it easy to retrieve more relevant information such as the device's coordinates. A-GPS (Assisted GPS) retrieval is also possible by using this method.

Device Requirements

This method is supported on Series 40 devices starting from Java Runtime 1.0.0 for Series 40. Symbian devices do not support this method.

For more information about the supported device models which use Java Runtime 1.0.0 for Series 40, please visit the device specifications on Nokia Developer web site here: Series 40 devices with Java Runtime 1.0.0

Determining whether a device supports cell ID location retrieval on runtime

Retrieving location via cell ID on a device that doesn't satisfy the above requirement, will throw a "NoClassDefFoundError" and terminate the MIDlet's execution. There are two ways to identify on runtime, whether a device supports this method of location retrieval or not:

- By catching the `NoClassDefFoundError` as shown below
- By retrieving the availability of the network, with the following System property:

```
System.getProperty("com.nokia.mid.networkavailability");
```

The above property, like the cell ID retrieval itself, is introduced in Java Runtime 1.0.0 for Series 40. Retrieving the property on Series 40 devices that do not support Java Runtime 1.0.0, will result in a null value, which is also an indication that the device doesn't support cell ID retrieval. What is more, the value of this property on devices that support Java Runtime 1.0.0 for Series 40, can inform about the availability of the network. If a network is available, a cell ID retrieval can be attempted.

These are the following possible return values of the above property and their meanings:

Return String Value	Actual Case
null	The device does not support neither Java Runtime 1.0.0 for Series 40 and therefore nor cell ID retrieval
available	The device supports Java Runtime 1.0.0 for Series 40 and therefore cell ID retrieval. There is network coverage
unavailable	The device supports Java Runtime 1.0.0 for Series 40 and therefore cell ID retrieval. There is no network coverage or no SIM Card is inserted

Determining retrieval method

Getting the device's current location via cell ID is done similarly to a GPS location retrieval by using a location provider object and extracting it's location object. The following retrieval methods can be used:

```
//Online cell ID and/or WLAN
int[] methods = {(Location.MTA_ASSISTED | Location.MTE_CELLID | Location.MTE_SHORTRANGE
```


(C) Copyright Nokia 2014. All rights reserved.

```
| Location.MTY_NETWORKBASED)};

//Assisted GPS
int[] methods = {(Location.MTA_ASSISTED | Location.MTE_SATELLTITE |
Location.MTY_TERMINALBASED)};

//Standalone GPS
int[] methods = {(Location.MTA_UNASSISTED |  Location.MTE_SATELLTITE |
Location.MTY_TERMINALBASED)};

//Offline cell ID
int[] methods = {(Location.MTA_UNASSISTED |  Location.MTE_CELLID |
Location.MTY_TERMINALBASED)};
```

The retrieval of the location provider is made via a call to the getLocationProvider() method of the LocationUtil class:

```
provider = LocationUtil.getLocationProvider(methods, null);
```

Differences between frequent Cell-ID and GPS based location updates

One important difference between a Cell-ID based and a GPS based location provider, is that it is not possible to set a Location Listener to the Cell-ID based provider for listening to frequent location updates as one could do with the GPS based provider. Since the same provider class is used in both Cell-ID and GPS location retrievals, the setLocationListener method can be called by objects that use these two different location retrieval methods. Even though this is syntactically correct, the setLocationListener method has not been implemented for Cell-ID based providers. The only way to update the user's current location in a Cell-ID based scenario, is to call the getLocation method within a repeating thread.

Implementing the main MIDlet

```
import com.nokia.mid.location.LocationUtil;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;
import javax.microedition.lcdui.Form;
import javax.microedition.location.LocationProvider;
import javax.microedition.location.Location;
import javax.microedition.location.LocationException;
import javax.microedition.location.QualifiedCoordinates;

public class GetCellIdCoordinates extends MIDlet implements Runnable, CommandListener
{
        LocationProvider provider;
        Location location;
        QualifiedCoordinates coordinates;
        Thread t;
        Form f;
        Display display;
        Command exitCommand=new Command("Exit", Command.EXIT,0);
    protected void startApp() throws MIDletStateChangeException
    {
        display = Display.getDisplay(this);
        f=new Form("Current Location!");
```


(C) Copyright Nokia 2014. All rights reserved.

```
        f.append("Getting current location by using Cell ID...");
        f.addCommand(exitCommand);
        f.setCommandListener(this);
        display.setCurrent(f);
        t=new Thread(this);
        t.start();
    }

    protected void destroyApp(boolean unconditional)
                    throws MIDletStateChangeException {
    }

    protected void pauseApp() {
    }

    public void run()
    {
        try {
            //Specify the retrieval method to Online/Cell-ID
            int[] methods = {(Location.MTA_ASSISTED | Location.MTE_CELLID |
Location.MTE_SHORTRANGE | Location.MTY_NETWORKBASED)};
            // Retrieve the location provider
            provider = LocationUtil.getLocationProvider(methods, null);
            location=provider.getLocation(50000);
            coordinates=location.getQualifiedCoordinates();
            f.deleteAll();
            f.append("Latitude:"+coordinates.getLatitude()+"\n");
            f.append("Longitude:"+coordinates.getLongitude());
        } catch (NoClassDefFoundError ex)
        {
            f.deleteAll();
            f.append("Cell-ID retrieval not supported on this device");

        } catch(InterruptedException e)
        {
            f.deleteAll();
            f.append("Location retrieval was interrupted");
        } catch(LocationException x)
        {
            f.deleteAll();
            f.append("Location could not be retreived");
        }

    }

    public void commandAction(Command c, Displayable d) {
        if(c==exitCommand)
        {
            try {
                destroyApp(false);
                notifyDestroyed();
                }
            catch (MIDletStateChangeException ex) {
            }
        }
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```
}
```

Resources

The source file as well as the binaries can be downloaded from here: File:GetCellIdCoordinates.zip

See also

- Java Developer's Library : Specifying the location provider

(C) Copyright Nokia 2014. All rights reserved.