How to use different positioning methods in S60

Introduction

There are different kind of positioning methods available in Symbian platform. This article briefly describes those and how to use each method from Symbian C++ application.

Description

Four different methods are available in Symbian platform. Each having different response time, accuracy and so on. Each location accusation method is identified with Uid and technology type. To use any of the method, we have to create a session with the positioning server and then we create a sub-session with the positioning server. When we create sub-session we can instruct the framework what positioning mechanism we are going to use. If we don't provide any specific positioning method then the server uses the positioning module with the highest priority by default. If the highest priority positioning module is not available or if it returns an error for a position request then the positioning module with the next highest priority is used. Following table shows different modules, name and it's Uid.

Technology Name	Used Uid
Network based	0x10206915
Assisted GPS	0x101fe98c
Integrated GPS	0x101fe98a
Bluetooth GPS	0x101fe999

Detecting numbers of available modules

Following code try to find the number of modules and identify the desired module id (set by UI) that is going to be used for positioning. In this example we use the position information to download map.

```
    TUint module = 0;
    // Get the available numbers of modules
    error = iPosServer.GetNumModules(module);
    if(error != KErrNone)
        {
        iPosServer.Close();
        return;
        }
    TPositionModuleInfo info;

    for(TUint i = 0; i < module; i++)
        {
        error = iPosServer.GetModuleInfoByIndex(i,info);
        TBool isavailable = info.IsAvailable();
        if(error == KErrNone && isavailable)
            {
            TBuf<200> buf(0);
            buf.SetLength(0);
            buf.FillZ();
            info.GetModuleName(buf);

            if(!buf.Compare(_L("Network based")) && iELocModuleUsed == eNetwork)
                {
                iTargetModule = info.ModuleId();
                break;
                }

            if(!buf.Compare(_L("Integrated GPS")) && iELocModuleUsed == eGPS)
                {
                iTargetModule = info.ModuleId();
                break;
                }

            if(!buf.Compare(_L("Assisted GPS")) && iELocModuleUsed == eAssisted)
                {
                iTargetModule = info.ModuleId();
                break;
                }

                if(info.TechnologyType()& TPositionModuleInfo::ETechnologyTerminal)
                    {
                        TPositionModuleInfo::TDeviceLocation deviceLoc =  info.DeviceLocation();
                        if (deviceLoc == TPositionModuleInfo::EDeviceInternal)
                            {
                            //Checked GPS is available, you can use the information to your app
                            }
                    }
            }
        else
            {
              //Checked GPS is available, you can use the information to your app
            }
        }
```

Using the desired module

Following code shows how to use the desired module to find the location. If the desired target type is set then that module id can be used to open a sub-

(C) Copyright Nokia 2013. All rights reserved.

session.

```
    // Open subsession to the position server if it was targeted as Network based
    if(iELocModuleUsed == eNetwork || iELocModuleUsed == eGPS || iELocModuleUsed == eAssisted)
        {
        if(iTargetModule != TUid::Null())
            {
            error = iPositioner.Open(iPosServer, iTargetModule);
            }
        else
            {
            // Open subsession to the position server if it was targetted as default
            error = iPositioner.Open(iPosServer);
            }
        }
    else
        {
        // Open subsession to the position server if it was targetted as default
        error = iPositioner.Open(iPosServer);
        }
```

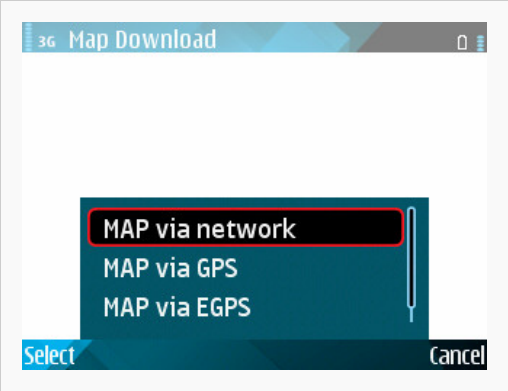
How to detect if a device support specific technology type

A device can support several technology type, that can be detected by application. Each technology can be enabled/disabled in the positioning settings of the devices. ETechnologyTerminal means the device has integrated GPS, ETechnologyNetwork means positioning technology is network based and so on. More information in LbsCommon.h in SDK.

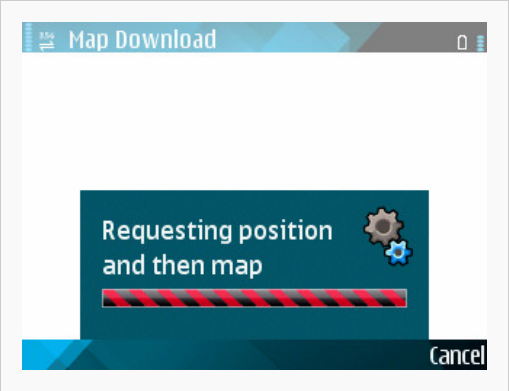
```
                if(info.TechnologyType()& TPositionModuleInfo::ETechnologyTerminal)
                    {
                        TPositionModuleInfo::TDeviceLocation deviceLoc =  info.DeviceLocation();
                        if (deviceLoc == TPositionModuleInfo::EDeviceInternal)
                            {
                            //Checked GPS is available, you can use the information to your app
                            }
                }
```

Using the Example

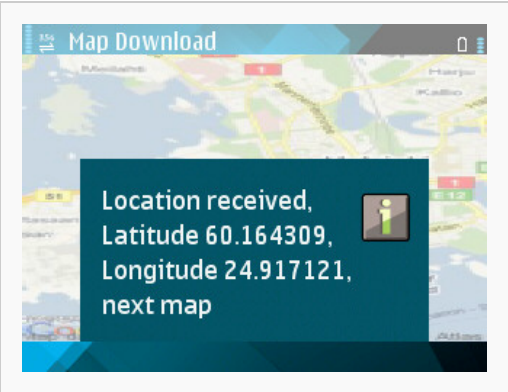
We can ask this example to get the location From UI and the location is used to download map. The obtained location is the center of the map. So first half of the operation is location accusation and second half is obtaining map from Google. When we use network based solution then we should not see the satellite icon (for example in N97) on the other hand when we use GPS (a relatively longer process) to acquire location then we should not see the network connection icon. In this example, we see the network connection icon when we download the map and use network based or assisted GPS.



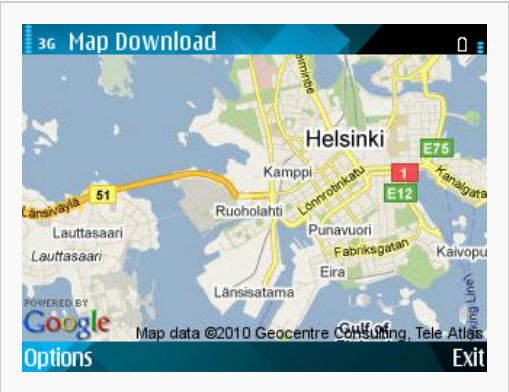
Select location technology type



Getting location



Location received



Map received

- Download the example: File:Locationmethod.zip
- Download the example for N8: File:N8LBSSample ModuleName.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.