Archived:How to get all LBS parameters in S60 3rd Edition FP1

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Overview

This example will give you the code needed to get all needed LBS data from WLAN, GPS and GSM

This is not a complete example but the files posted here: File:AllLBS.zip are ready to be used in any application. Just unzip the archive and use the example code below to add this engine to your application. It is that simple.

All 3 engines (WLAN, GPS and GSM) are implemented as Active Objects so they run asynchronously, in a safe way. Each time you request an information you get the latest information possible without waiting.

For WLAN, you get:

```
- all available Wireless networks
- BSSID
- SSID
- Power
- AP type
- Security type
```

For GSM:

```
- Power
- MCC
- MNC
- LAC
- Cell ID
- Long Name
```

For GPS:

```
- TPosition class wich gives altitude, latitude, longitude, time, speed, etc... look
SDK
```

Preconditions

This example has been only tested on Nokia N95. This example assumes that the target device has WLAN and GPS.

CAPABILITIES

This snippet requires capabilities. Self-signing is not possible because a Developer Certificate is needed. You can use Online Symbian-Signed.

ReadDeviceData ReadUserData NetworkServices Location LocalServices UserEnvironment WriteUserData WriteDeviceData

Libraries needed are specified in header files.

Network Engine

(C) Copyright Nokia 2014. All rights reserved.

This is the main class. Construct an object of type CNetworkEngine in your program to get full access to all the functions from GPS, WLAN and GSM.

If you want to get notifications from the network engine implement the observer and pass it as a parameter like in the example, else pass a NULL.

Here it is an example:

```
//In your .h file you've implemented the interface MNetworkEngineObserver like this
#include "NetworkEngine.h"
public YourClass: public MNetworkEngineObserver
{
        void YourFunctionL();

        /**
  * Functions used to handle events from NetworkEngine
  */
 void HandleErrorL(TInt aError);
 void HandleErrorL(TInt aError, const TDesC &aMessage);
 void HandleMessageL(const TDesC &aMessage);
}

//In your .cpp file
void YourFunctionL()
{
   CNetworkEngine *networkEngine;
   TRAPD(err, (networkEngine = CNetworkEngine::NewL(this)));
 if(err != KErrNone)
 {
  networkEngine = NULL;
  //print err message
                return;
 }

   CleanupStack::Push(networkEngine);

   TBuf<8192> bigBuf;

   //Get GSM Information (XML format)
   networkEngine->GetGSMInfoL(bigBuf);
   //print or do whatever with bigBuf

   //Get WLAN Information (XML format)
   networkEngine->GetWLANInfo(bigBuf);
   //print or do whatever with bigBuf

   //Get GPS Information (TPosition)
   TPosition pos;
   pos = networkEngine->GetGPSCoordonates();
   //do whatever with pos
}
```

Network Engine header file

```
/**
 * File NetworkEngine.h
```


(C) Copyright Nokia 2014. All rights reserved.

```
 *
 * Contains definitions for CNetworkEngine class and MNetworkEngineObserver interface
 *
 */

#ifndef __NetworkEngine_H__
#define __NetworkEngine_H__

#include "GSMInfo.h"
#include "wlaninfo.h"
#include "LBSInfo.h"

//Log entry size
const TInt KLogEntrySize = 32768;

// EventObserver interface
class MNetworkEngineObserver
{
public:
 virtual void HandleErrorL(TInt aError) = 0;
 virtual void HandleErrorL(TInt aError, const TDesC &aMessage) = 0;
 virtual void HandleMessageL(const TDesC &aMessage) = 0;
};

class CNetworkEngine : public MGSMInfoObserver, public MWLANInfoObserver, public
MPositionObserver
{
public:
 //From GSMInfo
 void HandleGSMInfoErrorL(TInt aError);
 void GetGSMInfoL(TDes &aNetworkInfo);

 //From WLANInfo
 void HandleWLANInfoErrorL(TInt aError);
 void HandleWLANInfoMessage(TInt aReason);
 void HandleWLANInfoMessage(const TDesC &aMessage);
 void GetWLANInfo(TDes& aWLANInfo);

 //From LBSInfo
 void LBSPositionUpdatedL(TPositionInfoBase& aPosInfo);
 void LBSErrorL(const TDesC& aErrorString);
 void LBSMessageL(const TDesC& aMessage);
 void LBSMessageL(TInt aCode, const TDesC& aMessage);
 TPosition GetGPSCoordonates();

 //Own functions
 static CNetworkEngine* NewLC(MNetworkEngineObserver* aObserver);
 static CNetworkEngine* NewL(MNetworkEngineObserver* aObserver);
 ~CNetworkEngine();

protected:
 void ConstructL(MNetworkEngineObserver* aObserver);
 CNetworkEngine();

private: //data
 /**
  * The GSMInfo objecet
```


(C) Copyright Nokia 2014. All rights reserved.

```
  */
  CGSMInfo *iGSMInfo;

  /**
   * The WLAN info object
   */
  CWLANInfo *iWLANInfo;

  /**
   * The LBSInfo object
   */
  CLBSInfo *iLBSInfo;

  /**
   * Event observer
   */
  MNetworkEngineObserver *iNetworkEngineObserver;


  /**
   * For Position Updates
   */
  TReal iLatitude, logLatitude;
  TReal iLongitude, logLongitude;
  TPosition iLastPosition;

 };

 #endif /*__NetworkEngine_H__*/
```

Source Files

All the source files for this example are in the archive attached: File:AllLBS.zip

Testing

As mentioned, this code has been testing in a Nokia N95 with very good results. It should work on any phone that supports S60 3rd edition FP1.

(C) Copyright Nokia 2014. All rights reserved.