Find Out Cell ID in 3rd Edition

To find out the phone's current cell ID in Symbian 3rd Edition devices, the application must be Symbian Signed, since it requires the ReadDeviceData capability. The programmer must use an active object because the calls are asynchronous. Below is an example of a class that retrieves the Cell ID by request.

Header Required:

```
#include<etel3rdparty.h>
```

Library Required:

```
LIBRARY etel3rdparty.lib
```

Capability Required:

```
Capability ReadDeviceData
```

NetworkInfo.h

```
#ifndef __NETWORKINFO_H__
#define __NETWORKINFO_H__

#include <etel3rdparty.h> // CTelephony

// Observer interface
class MNetworkInfoObserver
    {
    public:
        virtual void NetworkInfoRetrievedL(
            const CTelephony::TNetworkInfoV1& aNetworkInfo) = 0;
        virtual void HandleNetworkInfoError(TInt aError) = 0;
    };

// Active object to get network info
class CNetworkInfo : public CActive
    {
    public:
        static CNetworkInfo* NewL();
        ~CNetworkInfo();

        void GetNetworkInfoL(MNetworkInfoObserver* aObserver);

    protected:
        // from CActive
        void RunL();
        TInt RunError(TInt aError);
        void DoCancel();

    private:
        CNetworkInfo();
        void ConstructL();
```


(C) Copyright Nokia 2014. All rights reserved.

```
    private:
        CTelephony* iTelephony;
        CTelephony::TNetworkInfoV1 iNwInfo;
        CTelephony::TNetworkInfoV1Pckg iNwInfoPckg;
        MNetworkInfoObserver* iObserver;
    };

#endif // __NETWORKINFO_H__
```

NetworkInfo.cpp

```
#include "NetworkInfo.h"

CNetworkInfo::CNetworkInfo()
    : CActive(EPriorityStandard),
      iNwInfoPckg(iNwInfo)
    {
    CActiveScheduler::Add(this);
    }

CNetworkInfo* CNetworkInfo::NewL()
    {
    CNetworkInfo* self = new (ELeave) CNetworkInfo;
    CleanupStack::PushL(self);
    self->ConstructL();
    CleanupStack::Pop();
    return self;
    }

void CNetworkInfo::ConstructL()
    {
    iTelephony = CTelephony::NewL();
    }

CNetworkInfo::~CNetworkInfo()
    {
    Cancel();
    delete iTelephony;
    }

// This function is used by our class' users to start getting network info.
void CNetworkInfo::GetNetworkInfoL(MNetworkInfoObserver* aObserver)
    {
    __ASSERT_ALWAYS(!IsActive(), User::Leave(KErrInUse));
    iObserver = aObserver;

    // Start async call to receive current network information
    iTelephony->GetCurrentNetworkInfo(iStatus, iNwInfoPckg);
    SetActive();
    }

void CNetworkInfo::DoCancel()
    {
    iTelephony->CancelAsync(CTelephony::EGetCurrentNetworkInfoCancel);
    }
```


(C) Copyright Nokia 2014. All rights reserved.

```
void CNetworkInfo::RunL()
    {
    User::LeaveIfError(iStatus.Int());
    // Request completed successfully.
    // Now we can notify our observer.
    if(iObserver)
        {
        iObserver->NetworkInfoRetrievedL(iNwInfoPckg());
        }
    }

TInt CNetworkInfo::RunError(TInt aError)
    {
    // There was an error retrieving current network info.
    // Let's inform our observer about the error so that it can analyze it
    // and try to recover.
    if(iObserver)
        {
        iObserver->HandleNetworkInfoError(aError);
        }

    return KErrNone;
    }
```

Related Links

- Usage of CTelephony
- Cell ID with CTelephony

(C) Copyright Nokia 2014. All rights reserved.