



| | | | |
|-----------------|-----------------|--------------------------|-------------|
| ID | - | Creation date | 4. May 2009 |
| Platform | S60 3rd Edition | Tested on devices | E61i |
| Category | Symbian C++ | Subcategory | |

Keywords (APIs, classes, methods, functions): CTelephony::GetCurrentNetworkInfo()

Overview

CTelephonyReader implementation illustrates how to get MCC and MNC in S60 3rd Edition devices with CTelephony API. Basically IMSI is 15 digits long number. The first 3 digits of IMSI are the MCC, and is followed by the MNC, either 2 digits (European standard) or 3 digits (North American standard). So retriving MCC and MNC from IMSI is not always safe, thus implemented this method to get MCC MNC from CTelephony::GetCurrentNetworkInfo() API of symbian.

The implementation is pretty simple, and when using this one only thing to do in calling class is to implement the callback interface and then to construct an instance of the CTelephonyReader.

Headers Required

```
#include <Etel3rdParty.h>
```

Library Needed

```
LIBRARY etel3rdparty.lib
```

Capability Required

```
Capability ReadDeviceData
```

TelephonyReader.h

```
#ifndef __TELEPHONYREADER_h__
#define __TELEPHONYREADER_h__
```

Get_MCC_MNC_using_CTelephony

```
#include <Etel3rdParty.h>
class MTelephonyObserver
{
public:
virtual void GetMccMnc(const TDesC& aMcc,const TDesC& aMnc) = 0;

};

class CTelephonyReader : public CActive
{
public:
static CTelephonyReader* NewL(MTelephonyObserver* aObserver);
static CTelephonyReader* NewLC(MTelephonyObserver* aObserver);
void ConstructL(void);
~CTelephonyReader()
protected:
void DoCancel();
void RunL();

private:
CTelephonyReader(MTelephonyObserver* aObserver);

private:
MTelephonyObserver ; iObserver
CTelephony ; iTelephony
CTelephonyInfoV1 ; iMCCMNCV1
CTelephonyInfoV1Pkg iMCCMNCV1Pkg;

};

#endif //__TELEPHONYREADER_h__
```

TelephonyReader.cpp

```
#include "TelephonyReader.h"

CTelephonyReader* CTelephonyReader::NewL(MTelephonyObserver* aObserver)
{
CTelephonyReader* self = NewLC(aObserver);
CleanupStack::Pop(self);
return self;
}

CTelephonyReader* CTelephonyReader::NewLC(MTelephonyObserver* aObserver)
{
CTelephonyReader* self = new (ELeave) CTelephonyReader(aObserver);
CleanupStack::PushL(self);
self->ConstructL();
return self;
}

CTelephonyReader::CTelephonyReader(MTelephonyObserver* aObserver)
:CActive(0),iObserver(aObserver),iMCCMNCV1Pkg(iMCCMNCV1)
{
}

CTelephonyReader::~CTelephonyReader()
```

TelephonyReader.h

Get_MCC_MNC_using_CTelephony

```
{
    ()Cancel
}

void CTelephonyReader::ConstructL(void)
{
    CActiveScheduler::Add(this);

    iTelephony = CTelephony::NewL();

    iTelephony->GetCurrentNetworkInfo(iStatus, iMCCMNCV1Pkg);

    SetActive
}

void CTelephonyReader::DoCancel()
{
    iTelephony->Cancel(CTelephony::EGetCurrentNetworkInfoCancel);
}

void CTelephonyReader::RunL()
{
    if(iStatus == KErrNone)
    {
        iObj = CTelephonyReader::GetMccMnc(iMCCMNCV1.iCountryCode, iMCCMNCV1.iNetworkId);
    }
}
```

Example Code

- [Example code](#)