Archived:Get MCC MNC using CTelephony

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*}}.

This code example shows how to get the MCC and MNC using the Symbian C++ `CTelephony` API.

Overview

The `CTelephonyReader` class shown below uses the CTelephony API to get MCC and MNC in S60 3rd Edition devices. It does this by extracting them from the IMSI. The IMSI is 15 digits long - the first 3 digits are the MCC, and these are followed by the MNC, either 2 digits (European standard) or 3 digits (North American standard).

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__

#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;
 CTelephony::TNetworkInfoV1   iMCCMNCV1;
 CTelephony::TNetworkInfoV1Pckg iMCCMNCV1Pkg;

};

#endif //__TELEPHONYREADER_h__
```

TelephonyReader.cpp

```
#include "TelephonyReader.h

CTelephonyReader* CTelephonyReader::NewL(MTelephonyObserver* aObserver)
    {
    CTelephonyReader* self = NewLC(aObserver);
```


(C) Copyright Nokia 2013. All rights reserved.

```
    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()
{
 Cancel();
}

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

 iTelephony = CTelephony::NewL();

 iTelephony->GetCurrentNetworkInfo(iStatus,iMCCMNCV1Pkg);

 SetActive();
}

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


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

Example Code

- Example code

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.