

ID	TSS001440	Creation date	June 23, 2009
Platform	S60 3rd Edition, FP1 S60 3rd Edition, FP2	Devices	Tested on Nokia N96
Category	Symbian C++	Subcategory	

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

Description

The phone network registration status can be monitored with the `CTelephony::NotifyChange()` function, with `ENetworkRegistrationStatusChange` as the second parameter.

Solution

The following sample code explains how to monitor changes in the network registration status when the phone is registering to a network.

NetworkRegister.h

```
#include <e32base.h>
#include <etelmm.h> // link against etel.lib
#include <Etel3rdParty.h> // link against Etel3rdParty.lib

class CNetRegister : public CActive
{
public:
    // static constructor + destructor
    static CNetRegister* NewL();
    ~CNetRegister();

public:
    void IssueRequest();

protected: // from CActive
    virtual void DoCancel();
    virtual void RunL();
};
```

Description

TSS001440_-_Monitoring_the_network_registration_status_using_CTelephony

```
private:    // private constructors

    CNetRegister();
    void ConstructL();

private:    // data
    RTelServer          iTelServer;
    RMobilePhone        iPhone;
    TFileName           iTsyName;
    CTelephony* iTelephony;
    CTelephony::TNetworkRegistrationV1 iNetworkRegistrationV1;
    CTelephony::TNetworkRegistrationV1Pckg iNetworkRegistrationV1Pckg;
    RMobilePhone::TMobilePhoneNetworkMode iRegStatus;
};
```

NetworkRegister.cpp

```
#include "NetworkRegister.h"

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

CNetRegister::CNetRegister()
: CActive(EPriorityHigh),
  iNetworkRegistrationV1Pckg(iNetworkRegistrationV1)
{
}

void CNetRegister::ConstructL()
{
    iTelephony = CTelephony::NewL();
    User::LeaveIfError(iTelServer.Connect());

    // Load the appropriate TSY
    iTelServer.RTelServer::GetTsyName(0, iTsyName);
    User::LeaveIfError(iTelServer.LoadPhoneModule(iTsyName));

    // Open the first available phone
    RTelServer::TPhoneInfo phoneInfo;
    User::LeaveIfError(iTelServer.GetPhoneInfo(0, phoneInfo));
    User::LeaveIfError(iPhone.Open(iTelServer, phoneInfo.iName));
    CActiveScheduler::Add(this);
    IssueRequest();
}

CNetRegister::~CNetRegister()
{
    Cancel();
}
```

TSS001440_-_Monitoring_the_network_registration_status_using_CTelephony

```
delete iTelephony;
iTelServer.Close();
}

void CNetRegister::RunL()
{
    if( iStatus == KErrNone )
    {
        CTelephony::TRegistrationStatus regStatus = iNetworkRegistrationV1.iRegStatus;
        // regStatus now contains the current network registration status
    }
    else
    {
        // Handle errors
    }

    IssueRequest();
}

void CNetRegister::DoCancel()
{
    iTelephony->CancelAsync(CTelephony::ENetworkRegistrationStatusChangeCancel);
}

void CNetRegister::IssueRequest()
{
    iTelephony->NotifyChange( iStatus,
                             CTelephony::ENetworkRegistrationStatusChange,
                             iNetworkRegistrationV1Pckg);

    SetActive();
}
```

Possible values for network registration state are listed below:

ERegistrationUnknown	Registration status is unknown.
ENotRegisteredNoService	Not registered. The ME can not detect any other networks and is not currently searching for a new operator to register to.
ENotRegisteredEmergencyOnly	Not registered. The ME can detect other networks in which it is possible to make emergency calls only.
ENotRegisteredSearching	Not registered, but the ME is currently searching for a new operator to register to.
ERegisteredBusy	Registered, network busy.
ERegisteredOnHomeNetwork	Registered on home network.
ERegistrationDenied	Registration denied.
ERegisteredRoaming	Registered, roaming.

Required capabilities: ReadDeviceData.