Monitoring signal strength with CTelephony

`CNwSignalCheck` implementation illustrates how to check and monitor the signal strength of S60 3rd Edition devices with `CTelephony` API.

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 `CNwSignalCheck`.

When constructing `CNwSignalCheck`, the initial signal strength is checked with `GetSignalStrength()` -function and after this function returns and calls back the `RunL()`, all changes in the signal strength are monitored by calling `NotifyChange()` for the signal strength status.

Link against:

```
LIBRARY etel3rdparty.lib
```

SignalStrenght.cpp

```
#include "SignalStrenght.h"

CNwSignalCheck* CNwSignalCheck::NewLC(MNwSignalObserver& aObserver)
{
    CNwSignalCheck* self = new (ELeave) CNwSignalCheck(aObserver);
    CleanupStack::PushL(self);
    self->ConstructL();
    return self;
}

CNwSignalCheck* CNwSignalCheck::NewL(MNwSignalObserver& aObserver)
{
    CNwSignalCheck* self = CNwSignalCheck::NewLC(aObserver);
    CleanupStack::Pop(); // self;
    return self;
}

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


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


CNwSignalCheck::CNwSignalCheck(MNwSignalObserver& aObserver)
: CActive(EPriorityStandard),iObserver(aObserver),iSigStrengthV1Pckg(iSigStrengthV1)
{
    CActiveScheduler::Add(this);
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
void CNwSignalCheck::GetSignalInfo()
{
    if(iTelephony && !IsActive())
    {
        iGettingSignal = ETrue;
        iTelephony->GetSignalStrength(iStatus, iSigStrengthV1Pckg);
        SetActive();
    }
}

void CNwSignalCheck::RunL()
{
    iObserver.SignalStatus(iSigStrengthV1.iSignalStrength,iSigStrengthV1.iBar);

    if(iStatus != KErrCancel)
    {

iTelephony->NotifyChange(iStatus,CTelephony::ESignalStrengthChange,iSigStrengthV1Pckg);
        SetActive();
    }

    iGettingSignal = EFalse;
}

void CNwSignalCheck::DoCancel()
{
    if(iGettingSignal)
        iTelephony->CancelAsync(CTelephony::EGetSignalStrengthCancel);
    else
        iTelephony->CancelAsync(CTelephony::ESignalStrengthChangeCancel);
}
```

SignalStrenght.h

```
#include <Etel3rdParty.h>

class MNwSignalObserver
{
    public:
        virtual void SignalStatus(TInt32 aStrength,TInt8 aBars) = 0;
};


class CNwSignalCheck : public CActive
{
    public:
        ~CNwSignalCheck();
        static CNwSignalCheck* NewLC(MNwSignalObserver& aObserver);
        static CNwSignalCheck* NewL(MNwSignalObserver& aObserver);

    private:
        CNwSignalCheck(MNwSignalObserver& aObserver);
        void ConstructL(void);
```


(C) Copyright Nokia 2014. All rights reserved.

```
    private:
        void GetSignalInfo();
        void RunL();
        void DoCancel();

    private:
        MNwSignalObserver& iObserver;
        CTelephony* iTelephony;
        CTelephony::TSignalStrengthV1 iSigStrengthV1;
        CTelephony::TSignalStrengthV1Pckg iSigStrengthV1Pckg;
        TBool iGettingSignal;
};
```

Example Code

- Example code

(C) Copyright Nokia 2014. All rights reserved.