Obtaining current device position

The following code snippet demonstrates how to use class **CPositionReader** for obtaining current device position. This snippet also demonstrates, how to find the nearest landmark according to the current position.

Your class should implement interface **CPositionReaderObserver** and include instance of **CPositionReader** as a memeber.

It is necessary to include the following additional headers:

- *epos_landmarks.h*
- *epos_cposlmitemiterator.h*
- *epos_cposlandmarkdatabase.h*

Start async. request:

```
iReader = CPositionReader :: NewL( this );
iReader->ReadPosInfo();
```

You should add the following code in the method *ReadingComplete()* for reading current location and finding the nearest landmark:

```
// read result device position
TPositionInfo& info = ( TPositionInfo& )aPosInfo;
TPosition position;
info.GetPosition( position );

// open default landmark DB
CPosLandmarkDatabase* db = CPosLandmarkDatabase :: OpenL();
CleanupStack :: PushL( db );

// create iterator
CPosLmItemIterator* iter = db->LandmarkIteratorL();
CleanupStack :: PushL( iter );

// distance between device position and current landmark
TReal32 distance = KMaxTReal32;

TPosLmItemId itemId = iter->NextL(),  // current landmark ID
             resultItemId = KPosLmNullItemId; // nearest landmark ID
while( itemId != KPosLmNullItemId )
{
   // read current landmark
   CPosLandmark* lm = db->ReadLandmarkLC( itemId );

   // get current landmark position
   TPosition lmPosition;
   if( lm->GetPosition( lmPosition ) == KErrNone  )
   {
      TReal32 d;
      // check distance
      if( position.Distance( lmPosition, d ) == KErrNone && d < distance )
      {
         resultItemId = itemId;
         distance = d;
      }
   }

   // read next landmark ID
   itemId = iter->NextL();
   CleanupStack :: PopAndDestroy(); // lm
}

// landmark was found
if( resultItemId != KPosLmNullItemId )
{
   // read landmark from DB
   CPosLandmark* resultLm = db->ReadLandmarkLC( resultItemId );

   // read landmark name
   TPtrC lmName;
   resultLm->GetLandmarkName( lmName );

   // show nearest landmark name
   CAknInformationNote* informationNote;
   informationNote = new ( ELeave ) CAknInformationNote;
   informationNote->ExecuteLD( lmName );

   CleanupStack :: PopAndDestroy(); // resultLm
}

CleanupStack :: PopAndDestroy( 2 ); // iter db
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.