How to select and show a landmark

Standard Dialogs

Landmarks UI Selector API contains several classes that allow you to select landmarks as well as landmark categories.

Headers required:

```
#include <clmklandmarkselectordlg.h> //CLmkLandmarkSelectorDlg
#include <clmkcategoryselectordlg.h> //CLmkCategorySelectorDlg
#include <clmkeditordlg.h>  // CLmkEditorDlg
```

Library needed:

```
LIBRARY  LmkCommonUi.lib //CLmkLandmarkSelectorDlg , CLmkCategorySelectorDlg
```

Required Capabilities:

```
Capabilities LocalServices Location NetworkServices ReadUserData WriteUserData
```

The following code snippet demonstrates how to select one landmark

```
TLmkItemIdDbCombiInfo item;
CLmkLandmarkSelectorDlg* selectLandmarkDlg = CLmkLandmarkSelectorDlg :: NewL();
selectLandmarkDlg->SetMopParent( this );

// select one landmark
if( selectLandmarkDlg->ExecuteLD( item ) != 0 ) // dlg is deleted automatically
{
   // get item ID
   TPosLmItemId itemId  = item.GetItemId();

   // your code here

   delete item.GetLmDb(); // fixing memory leak
}
```

And next code snippet demonstrates how to select multiple landmarks:

```
CLmkLandmarkSelectorDlg* selectLandmarkDlg = CLmkLandmarkSelectorDlg::NewL();
selectLandmarkDlg->SetMopParent( this );
RArray< TLmkItemIdDbCombiInfo > arrItems;
CleanupClosePushL( arrItems );

// select list of active landmarks
if( selectLandmarkDlg->ExecuteLD( arrItems ) != 0 )
{
   // your code here
   delete arrItems[0].GetLmDb(); // fixing memory leak
}
CleanupStack::PopAndDestroy(); // arrItems
```

You can use class **CLmkCategorySelectorDlg** for selecting one or more categories in the same manner (include *clmkcategoryselectordlg.h*).

(C) Copyright Nokia 2014. All rights reserved.

Landmarks UI Add/Edit API allows to use standart dialog for view/edit the specified landmark. API contains one class **CLmkEditorDlg** (include *clmkeditordlg.h*).

The following code snippet demonstrates how to edit specified landmark:

```
// Opens the default landmark database
CPosLandmarkDatabase* db = CPosLandmarkDatabase :: OpenL( );
CleanupStack::PushL( db );

CLmkEditorDlg::TLmkEditorParams editParams;

// all attributes of the landmark will be presented
editParams.iAttributes = CLmkEditorDlg::ELmkAll;

// user can change values of the attributes
editParams.iEditorMode = CLmkEditorDlg::ELmkEditor;


// lmItem - ID of the specified landmark

CLmkEditorDlg *dlg = CLmkEditorDlg::NewL(*db, lmItem, editParams);
dlg->ExecuteLD();

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

Custom Dialogs

This example demostrates:

- how to select landmark from database
- how to read landmark data (latitude, longitude...)
- how to present landmark data

There are used following header files:

```
#include <clmklandmarkselectordlg.h>
#include <tlmkitemiddbcombiinfo.h>
#include <epos_cposlandmarkdatabase.h>
#include <e32cmn.h> //Abs
```

MMP-file must include following libraries:

```
LIBRARY lmkcommonui.lib
LIBRARY eposlandmarks.lib
LIBRARY lbs.lib
// for dialog
LIBRARY eikdlg.lib
LIBRARY eikctl.lib
```

It is necessary to note, that the values of coordinates of each landmark from the database are presented in decimal system - it is good for calculations. In pratice, for demonstrating latitude and longitude, there are used deg, minutes and seconds. Geographical latitudes of landmarks, located in the North Hemisphere are considered positive, and the latitude of landmarks in the South Hemisphere are considered negative. Geographical longtitude of marks, located west of Greenwich meridian are considered positive, and the longtitude of marks located east of Greenwich are considered negative.

For translating latitude/longitude from decimal system You can use the function `Decompose2Geographical`:

```
const TInt KMinsPerDegree = 60;
const TInt KSecsPerDegree = 3600;
```


(C) Copyright Nokia 2014. All rights reserved.

```
// decompose latitude/longitude in deg, minutes, seconds
void YourClassName :: Decompose2Geographical( TReal64& aValue, TInt& aDeg,
 TInt& aMinutes, TReal& aSeconds )
{
   aDeg     = ( TInt )aValue;
   aMinutes = ( Abs( aValue - aDeg ) ) * KMinsPerDegree;
   aSeconds = ( Abs( aValue - aDeg ) ) * KSecsPerDegree - aMinutes * KMinsPerDegree;
}
```

For presentation latitude/longitude as string, You can use following function:

```
// append latitude/longitude data in aDes
// i.e. if aValue = -23,3876 then aDes += -23°23'15.36"

void YourClassName :: AppendValueAsGeographical( TReal64& aValue,
 TDes& aDes )

{
   TInt deg, minutes;
   TReal secs;
   Decompose2Geographical( aValue, deg, minutes, secs );

   aDes.AppendNum( deg );
   aDes.Append( 176 ); // degree sign

   aDes.AppendNum( minutes );
   aDes.Append( '\'' );

   TRealFormat format( 5, 2 );
   format.iType = KRealFormatFixed | KDoNotUseTriads;

   aDes.AppendNum( secs, format );
   aDes.Append( '"' );
}
```

The choice of the landmark from the database and showing of its data are presented in the following snippet of code:

```
// this dialog allows to select landmark
CLmkLandmarkSelectorDlg* selectLandmarkDlg = CLmkLandmarkSelectorDlg :: NewL();
TLmkItemIdDbCombiInfo selectedItem;
// dlg is deleted automatically
TInt ret = selectLandmarkDlg->ExecuteLD( selectedItem );
if( ret != 0 )
{
   // Select or OK was pressed
   TPosLmItemId itemId = selectedItem.GetItemId();       // selected item ID
   CPosLandmarkDatabase* lmDb =  selectedItem.GetLmDb(); // selected item DB
   CPosLandmark* lmItemPos = lmDb->ReadLandmarkLC( itemId ); // read item data


   // !!! All dialogical strings must be defined in resources.
   // In this example (in order not to overflow the code) the strings are declared
   // as string literals (macro _LIT).
   TBuf<128> lmInfo;
   _LIT( KName, "Name: ");
   lmInfo.Append( KName );
   TPtrC landmarkName;
   if( lmItemPos->GetLandmarkName( landmarkName ) == KErrNone )
```


(C) Copyright Nokia 2014. All rights reserved.

```
      lmInfo.Append( landmarkName ); // landmark name is defined
   lmInfo.Append( '\n' );

   TLocality pos;
   if( lmItemPos->GetPosition( pos ) == KErrNone )
   {
      _LIT( KLatitude, "Latitude: " );
      lmInfo.Append(KLatitude);
      TReal64 latitude = pos.Latitude();
      AppendValueAsGeographical( latitude, lmInfo );
      lmInfo.Append( '\n' );

      _LIT( KLongitude, "Longitude: " );
      lmInfo.Append( KLongitude );
      TReal64 longitude = pos.Longitude();
      AppendValueAsGeographical( longitude, lmInfo );
      lmInfo.Append( '\n' );

      _LIT( KAltitude, "Altitude: " );
      lmInfo.Append( KAltitude );

      TRealFormat format( 6, 2 );

      lmInfo.AppendNum( pos.Altitude(), format );
      lmInfo.Append( '\n' );

      _LIT( KHorAcc, "Hor. Acc.: " );
      lmInfo.Append( KHorAcc );
      lmInfo.AppendNum( pos.HorizontalAccuracy(), format );
   }
   delete lmDb;

   // create and execute dialog
   CAknMessageQueryDialog* dlg = CAknMessageQueryDialog :: NewL( lmInfo );
   dlg->PrepareLC( R_LANDMARK_DIALOG ); // don't forget to include Your rsg-file
   _LIT( KDlgName, "Landmark Info" );
   dlg->SetHeaderTextL( KDlgName );
   dlg->RunLD();
}
```

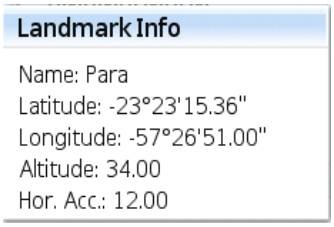
RSS-data for the dialog:

```
RESOURCE DIALOG r_landmark_dialog
{
   flags = EEikDialogFlagNoDrag | EEikDialogFlagCbaButtons | EEikDialogFlagWait;
   buttons = R_AVKON_SOFTKEYS_BACK;
   items =
   {
      DLG_LINE
      {
         type = EAknCtPopupHeadingPane;
         id = EAknMessageQueryHeaderId;
         control = AVKON_HEADING
         {
            headinglayout = R_AVKON_LIST_HEADING_PANE_POPUPS;
         };
      },
      DLG_LINE
      {
```


(C) Copyright Nokia 2014. All rights reserved.

```
            type = EAknCtMessageQuery;
            id = EAknMessageQueryContentId;
            control = AVKON_MESSAGE_QUERY
            {
               message="";
            };
        }
    };
}
```

The snapshot of the dialog:



Internal Links

- How to manage landmark categories
- KIS000784 - Landmarks category selector dialog leaks memory
- Landmarks/web client example using Carbide.c++ and UI designer
- How to use Landmarks API
- How to compact local landmark databases
- How to export landmarks from database to file
- How to import landmarks from file to database
- Execution of landmark operations
- How to obtain and save current location
- Retrieving location information

(C) Copyright Nokia 2014. All rights reserved.