All landmarks and categories are stored in databases, which may be local or remote. The primary identifier of the database is *URI*. The *URI* consists of a protocol specifier and the database location: "protocol://location".

Reviewer Approved 

For the local database:

- protocol is "*file*".
- location consists of disk drive and file name (with extension **.ldb*)

Example of the local database URI: "file://C:my_data.ldb"

You can use **Landmarks Database Management API** for the database management. The main class of this API is **CPosLmDatabaseManager**. (include *epos_cposlmdatabasemanager.h*, link against *eposlmdbmanlib.lib*)

A couple of small examples:

The following code snippet demonstrates how to rename default landmark database:

```
_LIT( KNewName, "New Database Name" );
const TInt KMaxDbNameLen = 64;
TBuf<KMaxDbNameLen> newName( KNewName );

// Main DB Manager;
CPosLmDatabaseManager* dbManager = CPosLmDatabaseManager :: NewL();
CleanupStack :: PushL( dbManager );

// Get default db URI
HBufC* defaultUri = dbManager->DefaultDatabaseUriLC();
if( defaultUri )
{
   // Get db info by URI
   HPosLmDatabaseInfo* dbInfo = HPosLmDatabaseInfo :: NewLC( *defaultUri );
   dbManager->GetDatabaseInfoL( *dbInfo );

   // Get settings from info
   TPosLmDatabaseSettings& settings = dbInfo->Settings();

   // Set new name and save settings
   settings.SetDatabaseName( newName );
   dbManager->ModifyDatabaseSettingsL( *defaultUri, settings );

   CleanupStack :: PopAndDestroy( 2 ); // dbInfo defaultUri
}
CleanupStack :: PopAndDestroy( 1 ); // dbManager
```

The following code snippet demonstrates how to copy all local databases from one drive to another:

```
// db manager;
CPosLmDatabaseManager* dbManager = CPosLmDatabaseManager :: NewL();
CleanupStack :: PushL( dbManager );

// protocol for the local DB
_LIT( KFileProto, "file" );
```

```
// array, that contains URI of the local DBs
CDesCArray* dbUriList = dbManager->ListDatabasesLC( KFileProto );

const TInt KMaxUriLen = 128;
TBuf<KMaxUriLen> uri, newUri;

TChar driveFrom = 'C',
      driveTo   = 'E';

TBuf<1> newDrive;
newDrive.Append( driveTo );

for(TInt i = 0; i < dbUriList->Count(); i++ )
{
   uri = (*dbUriList)[i];
   if( uri[7] == driveFrom )
   {
      // setup new URI: file://C... -> file://E...
      newUri = uri;
      newUri.Replace( 7, 1, newDrive );
      // check if such DB exists
      if( dbManager->DatabaseExistsL( newUri ) )
      {
         // delete if already exists
         dbManager->DeleteDatabaseL( newUri );
      }
      dbManager->CopyDatabaseL( uri, newUri );
   }
}

CleanupStack :: PopAndDestroy( 2 ); // iDbList dbManager
```

Internal Links

- How to export landmarks from database to file
- How to import landmarks from file to database