Adding a Contact Item

The following example shows how to create a new contact, add suitable fields, and add the contact to the database. In this example, three text fields with fieldtypes, text labels, and text values are added to a contact. Note that whenever a field is added to a contact, the contact becomes responsible for managing the memory and other resources associated with the field. Therefore, it is not necessary to delete these fields explicitly – popping them from the cleanup stack is sufficient.

Header file required:

```
#include <cntdb.h>
#include <cntitem.h>
#include <cntfldst.h>
```

`CContactDatabase` link against library `cntmodel.lib` , so add following line to your .mmp file.

```
LIBRARY   cntmodel.lib
```

Add following files in .cpp file.

```
// Some literal strings for the fields:

_LIT(KForenameLabel,"Forename");
_LIT(KSurnameLabel,"Surname");
_LIT(KWorkPhoneLabel,"Work Phone");
_LIT(KForename,"Satya");
_LIT(KOtherForename,"Naresh");
_LIT(KSurname,"Vattikuti");
_LIT(KWorkPhone,"+919985671193");

// Open the default contacts database:
CContactDatabase* contactsDb = CContactDatabase::OpenL();
CleanupStack::PushL(contactsDb);

// Create a contact card and add some fields:
CContactItem* contact = CContactCard::NewLC();
CContactItemField* field = CContactItemField::NewLC(KStorageTypeText, KUidContactFieldFamilyName);

field->SetMapping(KUidContactFieldVCardMapUnusedN);
field->SetLabelL(KSurnameLabel);
field->TextStorage()->SetTextL(KSurname);
contact->AddFieldL(*field);
CleanupStack::Pop();

field = CContactItemField::NewLC(KStorageTypeText, KUidContactFieldGivenName);
field->SetMapping(KUidContactFieldVCardMapUnusedN);
field->SetLabelL(KForenameLabel);
field->TextStorage()->SetTextL(KForename);
contact->AddFieldL(*field);
CleanupStack::Pop();

field = CContactItemField::NewLC(KStorageTypeText, KUidContactFieldPhoneNumber);
field->SetMapping(KUidContactFieldVCardMapTEL);
field->SetLabelL(KWorkPhoneLabel);
field->TextStorage()->SetTextL(KWorkPhone);
contact->AddFieldL(*field);
CleanupStack::Pop();

// Add the new contact to the database and set it as the own card:
contactsDb->AddNewContactL(*contact);
contactsDb->SetOwnCardL(*contact);

CleanupStack::PopAndDestroy(2); // contact contactsDb
```

The contact card that represents the electronic business card of the owner of the Symbian mobile phone can be set by calling the `CContactDatabase::SetOwnCardL()` function. Note that the CContactItemField::SetMapping() function is called for each of the fields created earlier. This function associates a field with one of the standard vCard fields and is required if the contact will be populated from or will generate a vCard. This mapping is also needed if the application developer wants the contact to be displayed in the Phonebook application.

Links

Reading and Editing a Contact Item

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.