Archived:Exporting a vCard item from contacts to a file using Symbian C++

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Overview

This snippet shows a simple function implementation to export one contact from the default contacts database as a vCard to a given file.

This snippet can be self-signed.

MMP file

The following capabilities and libraries are required:

```
CAPABILITY ReadUserData
```

```
LIBRARY euser.lib
LIBRARY estor.lib
LIBRARY efsrv.lib
LIBRARY cntmodel.lib
```

Source file

```
#include <e32cmn.h>   //TUid
#include <e32std.h>   //User
#include <e32base.h>  //CArrayPtr, CleanupStack
#include <e32def.h>   //TBool
#include <s32file.h>  //RFileReadStream
#include <f32file.h>  //RFs
#include <cntdb.h>    //CContactDatabase
#include <cntdef.h>   //CContactIdArray
#include <cntfilt.h>  //CCntFilter
```

```
TBool ExportVCardL(const TDesC& aFileName, TInt aItemIndex)
{
  RFs fileSession;
  RFile file;

  User::LeaveIfError(fileSession.Connect());
  CleanupClosePushL(fileSession);

  if (file.Replace(fileSession, aFileName, EFileWrite) != KErrNone)
      {
      //failed to create the file
      CleanupStack::PopAndDestroy(); //fileSession
      return EFalse;
      }
  CleanupClosePushL(file);

  //open a write stream to the file
  RFileWriteStream outputFileStream(file);
  CleanupClosePushL(outputFileStream);

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

  //create an array of contact IDs to export
  CContactIdArray* exportContacts = CContactIdArray::NewL();
  CleanupStack::PushL(exportContacts);

  //use a filter to get only contact items (e.g. templates are excluded)
  CCntFilter *filter = CCntFilter::NewLC();
  filter->SetContactFilterTypeALL(EFalse);
  filter->SetContactFilterTypeCard(ETrue);
  contactsDb->FilterDatabaseL(*filter);

  //create an array to hold all filtered contact items
  CContactIdArray* contactIds;
  contactIds = CContactIdArray::NewLC(filter->iIds);

  //add given contact(by index) to the array of contact IDs to export
  if((*contactIds).Count() >= aItemIndex)
      exportContacts->AddL((*contactIds)[aItemIndex] );

  CleanupStack::PopAndDestroy(2); //contactIds, filter

  //KVersitEntityUidVCard is used to identify a vCard
  TUid uid = TUid::Uid(KVersitEntityUidVCard);
  contactsDb->ExportSelectedContactsL(uid,
                                      *exportContacts,
                                       outputFileStream,
                                       //contact ID is no exported
                                       CContactDatabase::EExcludeUid);
```


(C) Copyright Nokia 2013. All rights reserved.

```
  CleanupStack::PopAndDestroy(5); //exportContacts,contactsDb,
                  //outputFileStream,file,fileSession

  return ETrue;
}
```

Postconditions

A contact given by the index from the default contacts database is exported as a vCard to the given file.

The function returns ETrue to the caller when the file creation succeeds or EFalse when the file creation fails.

See also

Archived:Importing a vCard item into the contacts database from a file using Symbian C++

(c) Copyright Nokia 2013. All rights reserved.

(c) Copyright Nokia 2013. All rights reserved.