Reading messages stored on SIM card using Symbian C++

This Symbian C++ code snippet shows how to read SMS messages stored in the SIM card.

Preconditions

The class should be derived from `MMsvSessionObserver`.

MMP file

```
// .mmp
...
CAPABILITY ReadUserData WriteUserData NetworkServices
...
LIBRARY msgs.lib // CMsvSession
LIBRARY smcm.lib // TSmsUtilities
LIBRARY gsmu.lib // CSmsPDU
...
```

Header file

```
// .h
...
#include <msvapi.h> // MMsvSessionObserver
...
protected: // MMsvSessionObserver
    void HandleSessionEventL(TMsvSessionEvent aEvent, TAny* aArg1, TAny* aArg2, TAny* aArg3);
...
```

Source file

```
// .cpp
...
#include <MTCLREG.h> // CClientMtmRegistry
#include <smsclnt.h> // CSmsClientMtm
#include <smscmds.h> // ESmsMtmCommandEnumeratePhoneStores
#include <TXTRICH.H> // CRichText
...
_LIT(KFileName, "e:\\simcardx.txt");
RFs& fs = CCoeEnv::Static()->FsSession();
TUint mode = EFileWrite+EFileStreamText+EFileShareAny;
RFileWriteStream file;
User::LeaveIfError(file.Replace(fs, KFileName, mode));
CleanupClosePushL(file);
file.WriteUint16L(0xFEFF); // Windows unicode file header.

    CMsvSession* msvs = CMsvSession::OpenAsyncL(*this);
    CleanupStack::PushL(msvs);
        CClientMtmRegistry* reg = CClientMtmRegistry::NewL(*msvs);
        CleanupStack::PushL(reg);
            CSmsClientMtm* mtm = static_cast<CSmsClientMtm*>(reg->NewMtmL(KUidMsgTypeSMS));
            CleanupStack::PushL(mtm);
                TMsvId imFolderId=0;
                {
                // Reads the SMS messages on the SIM and creates a copy of those
                // messages in an invisible folder under the SMS service in the
                // message store. If successful, the imFolderId member of the
                // operation's progress identifies the invisible folder which
                // contains the messages read from the SIM card.
                CMsvEntry* serviceEntry = msvs->GetEntryL(KMsvRootIndexEntryId);
                CleanupStack::PushL(serviceEntry);
                    TMsvId serviceId;
                    TSmsUtilities::ServiceIdL(*serviceEntry, serviceId, KUidMsgTypeSMS);
                    CMsvEntrySelection* selection = new (ELeave) CMsvEntrySelection();
                    CleanupStack::PushL(selection);
                        selection->AppendL(serviceId);
                        TBuf8<1> emp (KNullDesC8);
                        CMsvOperationActiveSchedulerWait* waiter = CMsvOperationActiveSchedulerWait::NewLC();
                            CMsvOperation* operation = mtm->InvokeAsyncFunctionL(ESmsMtmCommandEnumeratePhoneStores, *selection, emp
                            CleanupStack::PushL(operation);
                                waiter->Start();
                                User::LeaveIfError(waiter->iStatus.Int());
                                TSmsProgressBuf progressBuf;
                                progressBuf.Copy(operation->ProgressL());
                            CleanupStack::PopAndDestroy(operation);
                            TSmsProgress progress = progressBuf();
                            if (!progress.iError)
                                {
                                //identify the invisible folder that contains the messages read from the phone store(SIM)
                                simFolderId = progress.iEnumerateFolder;
                                }
                        CleanupStack::PopAndDestroy(waiter);
                    CleanupStack::PopAndDestroy(selection);
                CleanupStack::PopAndDestroy(serviceEntry);
                }
                {
                // Now iterate through all the SMS entries in the invisible folder
                // and output their info to a text file
                TMsvSelectionOrdering sort;
```


(C) Copyright Nokia 2013. All rights reserved.

```
                sort.SetShowInvisibleEntries(ETrue); // To handle invisible entries also
                CMsvEntry* inboxContext=CMsvEntry::NewL(*msvs,simFolderId ,sort);
                CleanupStack::PushL(inboxContext);
                    CMsvEntrySelection* entries = inboxContext->ChildrenL();
                    CleanupStack::PushL( entries );
                    TInt count=entries->Count();
                    for (TInt i=0; i<count; i++)
                        {
                        TMsvId entryID = entries->At(i);
                        mtm->SwitchCurrentEntryL(entryID);
                        CMsvEntry* entry= msvs->GetEntryL(entryID);
                        CleanupStack::PushL(entry);
                            TMsvEntry entry1 =  entry->Entry();
                            CMsvStore* inboxStore= entry->ReadStoreL();
                            CleanupStack::PushL(inboxStore);
                                // if this is a SMS message and not empty
                                if ((entry1.iMtm==KUidMsgTypeSMS)&&(inboxStore->HasBodyTextL()))
                                    {
                                    mtm->LoadMessageL();
                                    // the message body
                                    CRichText& richText= mtm->Body();
                                    inboxStore->RestoreBodyTextL(richText);
                                    const TInt length = richText.DocumentLength();
                                    TPtrC body = richText.Read(0,length);
                                    if(body.Length()>0)
                                        {
                                        _LIT(KComma, ",");
                                        _LIT(KCrLf, "\x0D\x0A");
                                        _LIT(KFormat, "%F%Y-%M-%D %H:%T:%S.%C");
                                        // the Recipient Number or its Contact's Name (if available)
                                        file.WriteL(entry1.iDetails);
                                        file.WriteL(KComma);

                                        // the Recipient Number
                                        TPtrC number = mtm->SmsHeader().FromAddress();
                                        file.WriteL(number);
                                        file.WriteL(KComma);

                                        // the received date
                                        TBuf<32> buf;
                                        entry1.iDate.FormatL(buf, KFormat);
                                        file.WriteL(buf);
                                        file.WriteL(KComma);

                                        file.WriteL(body);

                                        file.WriteL(KCrLf);
                                        }
                                    }
                            CleanupStack::PopAndDestroy(inboxStore);
                        CleanupStack::PopAndDestroy(entry);
                        }
                    CleanupStack::PopAndDestroy(entries);
                CleanupStack::PopAndDestroy(inboxContext);
                }
            CleanupStack::PopAndDestroy(mtm);
        CleanupStack::PopAndDestroy(reg);
    CleanupStack::PopAndDestroy(msvs);

CleanupStack::PopAndDestroy(&file);
```

Related article: Deleting messages stored on SIM card using Symbian C++

(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.