



ID	CS001415	Creation date	June 9, 2009
Platform	S60 3rd Edition S60 5th Edition	Tested on devices	Nokia 5800 XpressMusic
Category	Symbian C++	Subcategory	Messaging

Keywords (APIs, classes, methods, functions): CMsvSession, CMsvEntry, TMsvId

Overview

This snippet demonstrates how to delete an incoming SMS message from the inbox folder.

MMP file

The following libraries and capabilities are required:

```
CAPABILITY  ReadUserData WriteUserData UserEnvironment
LIBRARY      msgs.lib
LIBRARY      smcm.lib
LIBRARY      gsmu.lib
```

Header file

```
#include <e32base.h>
#include <msvapi.h>           // MMsVSessionObserver
#include <msvids.h>          // Folder Ids
#include <smut.h>            // KUIDMsgTypeSMS

class CYourSmsEngine :    public CBase
{
public:
    static CYourSmsEngine* NewL();
};
```

CS001415_-_Deleting_an_incoming_SMS_message

```
static CYourSmsEngine* NewLC();
virtual ~CYourSmsEngine();

private:
    void HandleSessionEventL(TMsvSessionEvent aEvent,
        TAny* aArg1, TAny* aArg2, TAny* aArg3);

private:
    void ConstructL();
    CYourSmsEngine();

private:
    // Message body
    TBuf<KSmsMessageLength> iMessage;

    // Address (phonenumber)
    TBuf<KAddressLength> iAddress;

    // Session with the messaging server
    CMsvSession* iMsvSession;

    // CMsvEntry accesses and acts upon a particular Message Server entry
    CMsvEntry* iMsvEntry;

    // Id of a new message
    TMsvId iNewMessageId;
};
```

Source file

```
#include "cyoursmsengine.h"

#ifdef __WINS__
    const TMsvId KObservedFolderId = KMsvDraftEntryId;
#else
    const TMsvId KObservedFolderId = KMsvGlobalInBoxIndexEntryId;
#endif

const TMsvId KInbox = KMsvGlobalInBoxIndexEntryId;

CYourSmsEngine* CYourSmsEngine::NewL()
{
    CYourSmsEngine* self = NewLC();
    CleanupStack::Pop(self);
    return self;
}

CYourSmsEngine* CYourSmsEngine::NewLC()
{
    CYourSmsEngine* self = new (ELeave) CYourSmsEngine();
    CleanupStack::PushL(self);
    self->ConstructL();
    return self;
}
```

CS001415_-_Deleting_an_incoming_SMS_message

```
void CYourSmsEngine::ConstructL()
{
    // SMS automatic receiving needs a session to the messaging server
    iMsvSession = CMsvSession::OpenAsyncL(*this);
}

CYourSmsEngine::CYourSmsEngine()
{
}

CYourSmsEngine::~CYourSmsEngine()
{
    delete iMsvEntry;

    if (iMsvSession)
        iMsvSession->Cancel();

    delete iMsvSession;
}

void CYourSmsEngine::HandleSessionEventL(
    TMsvSessionEvent aEvent, TAny* aArg1, TAny* aArg2, TAny* /*aArg3*/)
{
    switch (aEvent)
    {
        case EMsvServerReady:
        {
            // Initialise iMsvEntry
            if (!iMsvEntry)
            {
                iMsvEntry = CMsvEntry::NewL(*iMsvSession, KInbox,
                                             TMsvSelectionOrdering());
            }
            break;
        }
        case EMsvEntriesCreated:
        {
            // Only look for changes in the Inbox
            if (aArg2 && *(static_cast<TMsvId*>(aArg2)) == KObservedFolderId)
            {
                CMsvEntrySelection* entries =
                    static_cast<CMsvEntrySelection*>(aArg1);
                if( entries->Count() >= 1 )
                    iNewMessageId = entries->At(0);
                else
                    return;
            }
            break;
        }
        case EMsvEntriesChanged:
        {
            // Look for changes. When using the emulator
            // observed folder is drafts, otherwise inbox
            if (aArg2 && *(static_cast<TMsvId*>(aArg2)) == KObservedFolderId)
            {
                CMsvEntrySelection* entries =
                    static_cast<CMsvEntrySelection*>(aArg1);
                if(!entries && entries->Count() < 1 )
                {
                    return;
                }
            }
        }
    }
}
```

CS001415_-Deleting_an_incoming_SMS_message

```
else if (iNewMessageId == entries->At(0))
{
    if( !iMsvEntry )
        return;

    // Set entry context to the new message
    iMsvEntry->SetEntryL(iNewMessageId);

    // Check the type of the arrived message and
    // that the message is complete
    // Only SMS's are our concern
    if (iMsvEntry->Entry().iMtm != KUidMsgTypeSMS ||
        !iMsvEntry->Entry().Complete())
        return;

    // Delete message from this Inbox-folder
    iMsvEntry->DeleteL(iMsvEntry->EntryId());
}
}
break;
}
}
}
```

Postconditions

The incoming SMS message is deleted.

See also

[CS001381 - Listening for incoming SMS messages](#)