

## Contents

- 1  
Introduction
- 2 Prerequisite
- 3 Sending  
message
  - ◆ 3.1  
SMSByExe.h
  - ◆ 3.2  
SMSByExe.cpp
- 4 Receiving  
Message
  - ◆ 4.1  
SmsHandler.cpp
- 5 Related  
Links:

## Introduction

It is a common practice to send messages and read incoming messages from Messaging server and parse them accordingly. Developer generally prefers such a functionality as being server side implementation which can be developed using GUI-less Exe.

This article is about to implement such SMS sending/receiving functionalities in GUI-less Exe (Server).

## Prerequisite

- Create one GUI-less exe project( for e.g: **SMSByExe** ) from Application Wizard. Generally it is termed as Console-based Exe project in the Application Wizard.
- Download **SmsHandler.zip** as follows:
  - ◆ For S60 2nd edition = SMSHandler\_S60\_2nd.zip
  - ◆ For S60 3rd edition = SMSHandler\_S60\_3rd.zip
- Extracting **SmsHandler.zip** will result into **SmsHandler.h** and **SmsHandler.cpp**
- Copy-Paste **SmsHandler.h** into your project's **/inc** folder.
- Copy-Paste **SmsHandler.cpp** into your project's **/src** folder.
- Edit your **.mmp** file. Add an entry for **SmsHandler.cpp** in **SOURCE** directive.

SOURCE SMSHandler.cpp

- Edit your **.mmp** file. Add libraries for SMS handling, and capabilities required.

```
//Libraries included for SMS support-  
LIBRARY msgs.      lib smcm.lib gsmu.lib mtur.lib  
CAPABILITY ReadUserData WriteUserData NetworkServices
```

## Sending message

### SMSByExe.h

```
#ifndef __SMSBYEXE_H__
#define __SMSBYEXE_H__

// Include Files

#include <e32base.h>

class CSmsHandler; //forward declaration

// Function Prototypes

GLDEF_C TInt E32Main();

LOCAL_C void SendSMSL();

CSmsHandler* iSmsHandler;

#endif // __SMSBYEXE_H__

    • Open your SMSByExe.cpp file.
    • Include SmsHandler.h.

#include "SMSHandler.h" //Added for SMS Handling
```

### SMSByExe.cpp

The following code snippet illustrates how to initialize SMSHandler class.

```
// Include Files

#include "SmsByExe.h"
#include <e32base.h>
#include <e32std.h>
#include "SMSHandler.h"

// Local Functions
LOCAL_C void MainL()
{
    //
    // add your program code here, example code below
    //
    //This is for initializing SMSHandler
    iSmsHandler = CSmsHandler::NewL();
    SendSMSL();
}

LOCAL_C void DoStartL()
{
    // Create active scheduler (to run active objects)
    CActiveScheduler* scheduler = new (ELeave) CActiveScheduler();
```

## Sending-Receiving\_SMS\_through\_an\_Exe\_(Server)

```
CleanupStack::PushL(scheduler);
 CActiveScheduler::Install(scheduler);

MainL();

 CActiveScheduler::Start();

// Delete active scheduler
CleanupStack::PopAndDestroy(scheduler);
}

LOCAL_C void SendSMSL()
{
    <128788>SMSText,PhoneNumber;
    SMSByExeL("SMS By An EXE");
    // Replace the number you wish to send message
    PhoneNumberL("9999999999");
    iSmsHandler( PhoneNumber, SMSText );
}

// Global Functions
GLDEF_C TInt E32Main()
{
    // Create cleanup stack
    __UHEAP_MARK;
    CTrapCleanup* cleanup = CTrapCleanup::New();

    // Run application code inside TRAP harness, wait keypress when terminated
    TRAPD(mainError, DoStartL());

    delete cleanup;
    __UHEAP_MARKEND;
    return KErrNone;
}
```

## Receiving Message

- Open your **SmsHandler.cpp** file which is supplied from the .zip file.

### SmsHandler.cpp

```
void CSmsHandler::MessageReceivedL( TMsvId aEntryId )
{
    CMsvEntryEntry = iSession->GetEntryL( aEntryId ); // current entry
    CleanupStack serverEntry );
    TMsvEntry entry = serverEntry->Entry(); // currently handled message entry

    SetNewlyETrue );
    SetUnread( ETrue );
    SetMovable( ETrue );

    serverChangeL( entry ); // commit changes

//Added to retrieve message body
    // iDescription will have only first 32 characters from the message
```

## Sending-Receiving\_SMS\_through\_an\_Exe\_(Server)

```
const TDesC& descp = entry.iDescription;
<40FBM688>MessageArrived;
    MessageCopy(descp);

//Added to retrieve Phone Number of the Sender
->iSmsMtmCurrentEntryL(aEntryId);
->iSmsMtmMessageL();
    CSmsHeader = iSmsMtm->SmsHeader();

    TPtHeader.FromAddress();
const TDesC& phoneNumber = from;

    CleanupStackAndDestroy( serverEntry );
}
```

## Related Links:

- [Sending SMS with RSendAs](#)
- [SMS Utilities API](#)
- [SMS Receiver](#)
- [Reading SMS from Inbox](#)
- [Sending SMS in S60 3rd Edition - MTM](#)
- [SMS Operations](#)
- [How to send an SMS using sockets](#)
- [Create Local SMS](#)
- [How to Open SMS or MMS Editor](#)