SIM TOOL KIT

APPLICATION PROGRAMMING INTERFACE
version 3.1

Manual for System Integrator and Software Developer

MOBITEK System Sdn .Bhd. (207015-D)
6th Floor, Suite 16, IOI Business Park, Persiaran Puchong Jaya Selatan, Bandar Puchong Jaya, Puchong 47100, Selangor, Malaysia.
Tel: 03-80644288 Fax: 03-80642109 Web: www.mobitek.com.my E-mail: mobitek2007@mobitek.com.my

Table of Contents

COPYRIGHT

DISCLAIMER

MOBITEK makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, MOBITEK reserves the right to revise this publication and to make changes from time to time in the contents hereof without obligation to notify any person of such revision or changes. Trademarks and Registered Trademarks Products and product names mentioned in this document may be trademarks or registered trademarks of their respective owners.

REVISION HISTORY

EDITION	***ISSUED DATE***	***DESCRIPTION***
1st	1st of April, 2006	Draft release
2nd	8th of April, 2006	Added SMSSentDisplay()
3rd	15th of June, 2006	Release version 1.3
4th	24th of July, 2006	Remove SMS API from manual
5th	27th of September, 2006	Release version 1.4
6th	26th of November, 2006	APPENDIX 1: Using COM Components from Visual Studo .Net Directly added
7th	30th of July, 2007	Minor modification
8th	3rd of September, 2007	Version 2.0 released
9th	31st of October, 2007	Version 2.1 released
10th	20th of December, 2007	Version 2.3 released
11th	27th of February, 2008	FAQ section added
12th	15th of March, 2008	Version 3.0 released
13th	3rd of April, 2008	Version 3.1 released

INTRODUCTION

API TYPE: ActiveX DLL component (Component Object Model) for Windows

ActiveX Name: MobitekSTK3.dll

Version: 3.1

The API contains 3 classes:

- SIMToolKit – SIM Tool Kit
- SMS – Short Message Service
- USSD – Unstructured Supplementary Service Data
- Phonebook – Phonebook that stores name and number

Scope of Manual: This manual only covers the ***SIMToolKit*** class. For information on the ***SMS, USSD and Phonebook*** classes, please refer to the manual, *"SMS API v5 Manual.chm"*.

Pre-requisite:

1. System integrator, and software developer must posses programming skill, and knowledge in making reference to DLL file.
2. Any programming language that can use ActiveX DLL or COM such as:
   2.1 Visual Basic
   2.2 Visual Basic .Net
   2.3 Visual C++
   2.4 Visual Studio .Net
   2.5 Cold Fusion
   2.6 Any programming language that can call MobitekSTK2.dll
3. Operating System: Windows 98, 2000, XP, and 2003 (Vista not tested nor supported)

Features: Software developer can develop their own user interface or software application that can access and interact with the SIM menu (menu in the SIM card). System integrator can integrate business system or external programme that can access the SIM menu, and to interact with the SIM menu.

What STK API can do:

- to build an application to automate the reloading of prepaid account
- to build an application to automate *Maxis e-load*, and *Digi Flexi e-load*

What STK API cannot do:

- cannot store any programme into SIM
- cannot change, modify, edit, or delete the menu in the SIM

UPGRADING FROM VERSION 1 TO 2

If you are upgrading from version 1 to 2, then you may need to re-write your code. The following table will guide you on methods or properties that has changed:

	STK API version 1	***STK API version 2***
STK class		Same as version 1. You do not need to re-write the code when using STK.
Initialise GSM Modem	InitModem (COM port number, optional SIM PIN)	ModemInit (COM port number, optional SIM PIN)
Send SMS	SendSMSWC2(mobile number, outgoing message)	SendSMS(mobile number, outgoing message)
Disconnect GSM Modem	CloseModem()	ModemClose()

UPGRADING FROM VERSION 2.0 TO 2.1

1. New method in version 2.1 – **DeactivateSTK()**
   1.1. **DeactivateSTK()** will deactivate SIM Tool Kit.
   1.2. The differences between **CloseSTK()** and **DeactivateSTK()** are
      a) with **CloseSTK()**, the session is closed or stopped but STK is still active. You can still call **MainMenuDisplay()**;
      b) use **CloseSTK()** when you want to cancel a reload transaction;
      c) with **DeactivateSTK()**, STK session is terminated, you must call **ActivateSTK()** before calling **MainMenuDisplay()**;
      d) call **DeactivateSTK()** before you call methods and properties in *SMS, USSD, Phonebook classes*. Ie. you must terminate the STK session before you can send SMS, read SMS, issue USSD command, and access Phonebook.

NEW FEATURES IN VERSION 2.3

1. There are no new method nor property.
2. Better handling of *Digi Flexi e-load* that returns non-standard response.
3. Automatically re-configure the STK modem in order to send and receive SMS.
   a) The response time to obtain "True" or "False" for the **MainMenuDisplay()** will be longer in version 2.3 than in 2.1 as the function will automatically re-configure the STK modem. However, the re-configuration process will occur once only.

NEW FEATURES IN VERSION 3.0

1. This version only works with STK modem – **Wavecom Fastrack Supreme 10, Type S10-STK3**. Other types of STK modem will not work.

2. There are no new method nor property in *STK class*.
3. *Phonebook class* added.
4. *SMS class* has a new method – **GetSignalStrength()**. Refer to the SMS API Manual for more information.
5. Method **MainMenuDisplay()** improved. Auto-reconfiguration of modem is only required when necessary. Wait for STK's response then only display items in main menu.

NEW FEATURES IN VERSION 3.1

1. This version only works with STK modem – **Wavecom Fastrack Supreme 10, Type S10-STK3**. Other types of STK modem will not work.
2. Method **MainMenuSelect()** improved. Previously all 1st try will get "MainMenuSelect() = False".

INSTALLATION

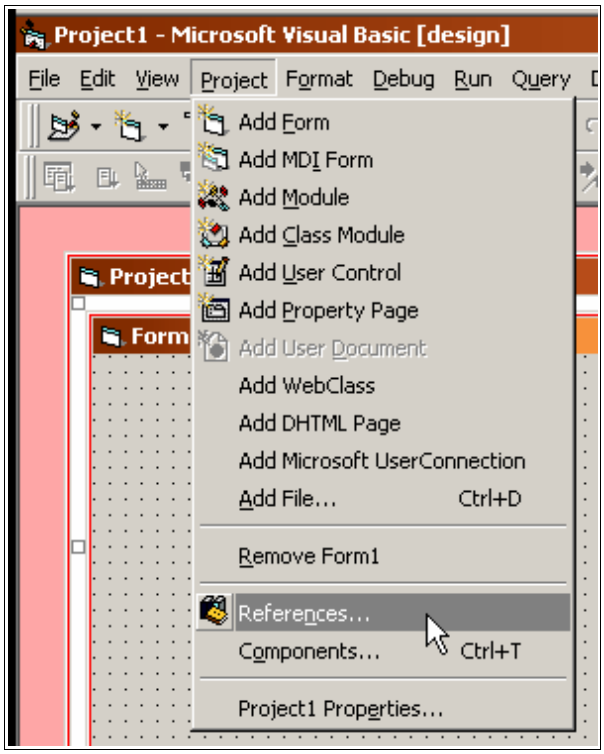
Run the “Setup.exe” to install “MobitekSTK2.dll” file. The dll file will be installed into “system32” folder”.

Before using the STK API (ActiveX), please go through the following check list:

1. GSM modem is properly set-up.
2. Identify the COM port where the GSM modem is connected.
3. The IDE (VB6, VB.net, VC++, etc.) is properly configured to use the ActiveX (refer to your programming guide)

Below is a Visual Basic example on how to configure to use the "MobitekSTK2.dll":

Goto “Project > References > MobitekSTK2”



SECTION 1: API FUNCTIONS

The 1st important step is to call method **ModemInit()** and obtain a return value of "1". This is to establish connection between server (PC) and STK modem. The **ModemInit()** is contained in the SMS class.

The **STKAPI.SIMToolKit** contain 12 methods:

(1) ActivateSTK() as Boolean

a)To turn on the SIM Tool Kit (STK). This is the 2nd important step after **ModemInit()**. If you do not call this method, then all other methods will fail to work.

b)Return value is:

= *TRUE* if activated

= *FALSE* if not

(2) CloseSTK() as Boolean

a)To close, cancel or stop a STK transaction. It does not terminate the STK session.

b)Return value is:

= *TRUE* if successfully ended

= *FALSE* if not

Tip: use this method when you want to cancel a reload transaction.

(3) DeactivateSTK () as Boolean

a)To terminate or exit the STK session. You must call this method before calling **SendSMS()** or **ReadSMS()**, otherwise you cannot send nor read SMS.

b)Call this method before using methods and properties in *SMS, USSD, Phonebook classes*.

c)You must terminate the STK session before you can send SMS, read SMS, issue USSD command, and access Phonebook.

d)Return value is:

= *TRUE* if successful

= *FALSE* if not

Tip: call this method before performing the following tasks:

- ***send SMS and read SMS***
- ***close your application***
- ***shut down the modem***
- ***shut down the server***

(4) MainMenuDisplay (*MainMenuName As String, MainMenuList As String*) as Boolean

a)To get main menu from STK. If return value is true, then the following values are retrieved:

- MainMenuName
- MainMenuList

b)Call this function only if

STKStatus = STKMainMenu

or

ActivateSTK() = True

(5) MainMenuSelect(*ByVal ItemSelect As Integer*) As Boolean

a)To select an item in main menu of STK.

b)Return value is:

= *True* if successfully submitted.

= *False* if not.

(6) SubMenuDisplay(*SubMenuName As String, SubMenuList As String*) As Boolean

a)To get sub menu from STK. If return value is true, then the following values are retrieved:

- SubMenuName
- SubMenuList

b)Call this function only if STKStatus = STKSubMenu

(7) SubMenuSelect(*ByVal ItemSelect As Integer*) As Boolean

a)To select an item in sub menu of STK.

b)Return value is:

= *True* if successfully submitted.

= *False* if not

(8) InputRequestDisplay() As String

a)To display the request by STK for user input.

b)Call this function only if STKStatus = STKRequestInput

(9) InputSubmit(ByVal UserInputSubmit As String) As Boolean

a)To submit user input to STK. The input can be alphanumeric or **"OK"**.

b)Use this only after calling InputRequestDisplay().

c)Return value is:

= ***TRUE*** if the submission is successful.

= ***FALSE*** if not.

(10) ResponseDisplay() As String

a)To display STK response after user has successfully submitted an input.

b)Call this function only if STKStatus = STKResponse

(11) SMSSentDisplay() As String

a)To display the outgoing SMS sent by STK.

b)Call this function only if STKStatus = STKSMSSent

(12) STKStatusGet() as STKStatus

a)To get the status of STK.

b)Return values is an integer representing status. The description of each status is as per STKStatus describe below:

STKMainMenu = 0 (STK has a main menu to show to user)

STKResponse = 1 (STK has a response to user's input)

STKInputRequest = 3 (STK has a request to user for input)

STKSubMenu = 6 (STK has a sub menu to show to user)

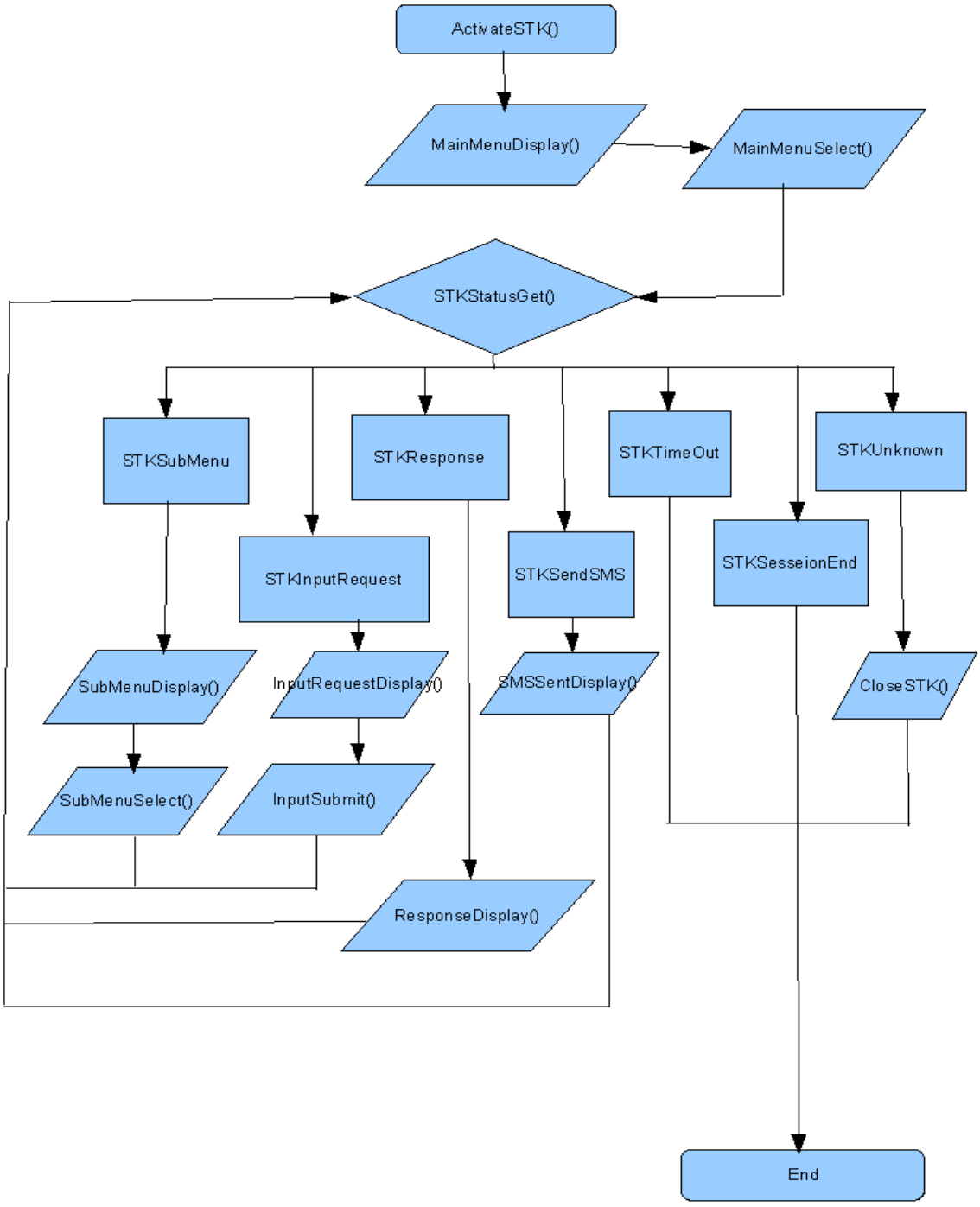
STKSMSSent = 9 (STK has sent out a SMS)

STKTimeOut = 98 (STK time out)

STKSessionEnd = 99 (STK session ends)

STKUnknown = 100 (Unknown status)

SECTION 2: STK API FLOW CHART



STK API must be used strictly according to the above flow. If the above flow is not followed, then there will be errors, or incomplete process will occur. If this happen, SI/SD must call StopSTK(), and ActivateSTK(), and restart the whole process again from the beginning.

SECTION 3: VISUAL BASIC SAMPLE

Please refer to the CD for VB sample code that uses STK API.

SECTION 4: DEPLOYMENT

When you want to deploy your STK application on another PC, you need to run the set-up file (***setup.exe***) for ***MobitekSTK3.dll*** which is located in the CD folder. Or you can use a Installer Maker (e.g. Wise Installer) to include the setup.exe with your application.

SECTION 5: TERMS AND CONDITIONS

5.1 ***STK API*** is **NOT free**. An annual license fee is payable.

5.2 ***STK API*** must be used together with the specified STK Modem supplied. Not all STK modems are able to support STK API.

SECTION 6: WARRANTY AND SUPPORT

6.1 With annual license fee, the MOBITEK System will provide **12 months** warranty, covering all the methods (***as stated in Section 1***) of the STK API that fail to work.

6.2 If a method fails, then a upgrade or patch file will be developed within 30 working days. The patch file will be send via e-mail.

6.3 Warranty does not cover customisation, adding new methods, and features to the STK API.

6.4 Support does not cover on-site support, nor telephone support.

6.5 Support shall be in the form of e-mail support, and response time shall be within 3 working days. All e-mail must contain the following information:

- Company name
- Invoice number
- Description of problem

and send it to
support2007@mobitek.com.my

APPENDIX 1: Using COM Components from Visual Studio .Net Directly

The following article is from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/callcomcomp.asp. It describes how you can call **ActiveX COM** from **.Net**:

As a .NET developer, you also have the option of using COM components directly. At least, that's what it looks like, although you're programmatic ally still using a RCW to get to objects in unmanaged code. If you're working within a Visual Basic .NET project, you can follow these steps to add a reference to a COM component:

1. *Click **Project**, and then click **Add Reference**.*
2. *In the **Add Reference** dialogue box, click the **COM** tab.*
3. *Select the type library you wish to use from the list and click **Select**, or use the **Browse** button to locate a component that's not listed. The selected components will be added to the lower listview in the dialogue box.*
4. *Click **OK** to create RCWs for the selected type libraries in your Visual Basic .NET project.*

When you do this, you'll find that Visual Basic .NET actually creates a DLL in your project's /Bin folder, with a name derived from the original COM component name. For example, if you reference BackEnd.dll version 2.0 in this manner, Visual Basic .NET will create the RCW in the file Interop.BackEnd_2_0.dll.

APPENDIX 2: FREQUENTLY ASKED QUESTIONS

Last updated on 27th of February, 2008.

Q: What is the difference between GSM modem and STK modem?

A: We use the term "STK modem" to refer to a GSM modem using STK API.

Q: When I call **ModemInit()**, and obtain a response of "SIMError". Why?

A: If the brand of STK modem is *iTegno*, then, sometimes, it may take 2-3 times to successfully initialise the STK modem. Ie. you need to re-execute the method **ModemInit()** a few times.

Q: In the middle of my process, I want to reset the STK modem, what is the correct flow?

A: The recommended flow is – half way through your code > **StopSTK()** > **DeactivateSTK()** > **ModemReset()** > **ModemClose()** > **ModemInit()** > **ActivateSTK()**

The wrong flow is – half way through your code > **ModemInit()**

If you do so, most likely you will get "**ModemInit() = SIMError**"

Q: Why do I receive a pop-up message – "Invalid GSM Modem connected!"?

A: You are using a STK modem that is not compatible with STK API. You must use STK modem supplied by MOBITEK System. Other types of modem do not work.

Q: Why do I get **"SendSMS() = False"** and **"ReadSMS() = False"** even though **"ModemInit() = 1"**?

A: The correct flow is – **ModemInit()** > code running STK process > **DeactivateSTK()** > **SendSMS()** or **ReadSMS()** > **ActivateSTK()** > code running STK process

The wrong flow is – **ModemInit()** > code running STK process > **SendSMS()** or **ReadSMS()**

Ie. you must deactivate the STK session before sending or reading SMS, otherwise you will get "**SendSMS() = False**" or "**ReadSMS() = False**"