The phone developer's guide for .Net

Revision: v1.1

This document only guide developers how to use the SDK. If you want to know the technical details, please refer to the class library.

We assume that the reader has already some .Net Development Foundation. This document will not make specific explanation about the specific .Net term and technology.

1. Base knowledge requirement

In development, developers need to have some knowledge of the method, property, event, the concept of the collection , singleton design mode and so on.

we recommend that initialize the instance of telephone and short message class by the GetInstance() method to ensure that the global memory is unique.

Reference information:

1、http://msdn.microsoft.com/zh-cn/library/ms173171.aspx
2、http://msdn.microsoft.com/zh-cn/library/ms173171(v=vs.80).aspx
3、http://msdn.microsoft.com/zh-cn/library/awbftdfh.aspx
4、http://msdn.microsoft.com/zh-cn/library/17sde2xt(v=vs.80).aspx
5、http://msdn.microsoft.com/zh-cn/library/system.collections.ilist.aspx

2. System architecture

3. Class diagram

None

4. Summary of design ideas

In this SDK, involving the method of operation of the hardware, unified with the static modifier, do not need to instantiate the class object. The application accepts the state change of SDK layer by Net event mechanism.

The persistence of short messages, Phone book, call records saved through the database. The application access such data through Collection.

As the characteristics of call record, we can do read and delete operation, but can not add operation. This type of operation is limit by the Collection.

5. Programming recommendation

5.1 When the related functions need to be used

1) Initialization module

a) Call SEUIC.Phone.Initialize.Init () or reload the specified function module.

b) Confirm whether the function call is successful,if not successful,call or reset the machine.

2) Anti-initialization

a) Ensure to call SEUIC.Phone.Initialize.UnInit() when ending the process.Especially, when the program is quiting unexpectedly, this method can be executed.

Note:If the method is not executed after the end of the process ,some thread will continue to execute according to the different initialization module.The reason is that the Win32API is in the critical region, the resources can not be cleaned up properly when the program quit unexpectedly.

3) Function call of each module

a) phone

b) short message

c)dial up

d) Module and network information

5.2 When do not need to use the telephone short message module

Please call SEUIC.Phone.Initialize.Init () method to reload the specified module.

Note: For disabled module or reset module, Please call SEUIC.Phone.Initialize.Init() method to initialize.

5.3 Development considerations

- Do not send short message when the system is in the phone status.You can query the current status of module before sending the short message.For more information please refer to 6.4.6
- Synchronous operation of the sending short message may take longer.Please try to call asynchronously by thread to avoid the interface is not responding.
- Send AT command. In order to avoid adverse impact caused by an error call, version is no longer provided to external after August 2, 2011.For more information please refer to 6.4.16
- In a non-call status(oduleStatus.STATE_CALL)), answer the call and hang up the method call is invalid, the system no longer accept the command.For more information please refer to 6.1.3、6.1.4

6. Each module development guide

6.1 Telephone

6.1.1 Call

Static method, use is as follows:

```
Call.MakeCall(sCallNumber);
```

The return value is CallResult enumeration.

Return CallResult.RIL_RESULT_OK if successful

Return CallResult.RIL_RESULT_ERROR if failed

The other state of the enumeration is temporarily unused.

The module will start dialing process after the success of dial_up.In this process,module will report the state and trigger the corresponding event.Please refer to the class library description.

6.1.2 Dial an extension (DTMF)

Static method, use is as follows:

```
Call.SendDTMF(string sNumber);
```

Return bool value,indicating that whether the method call is successful.If successful, send DTMF.

6.1.3 Answer

Static method, use is as follows:

```
Call.Answer();
```

Return bool value,indicating that whether the method call is successful.If successful,then answer to connect to the phone.

6.1.4 Hang up

Static method, use is as follows:

```
Call. HangUp ();
```

Return bool value,indicating that whether the method call is successful.If successful,then hang up.

6.1.5 Speaker on / off

Static method, use is as follows:

Turn on the speaker

```
Call.Speaker(true);
```

Turn off the speaker

```
Call.Speaker(true);
```

6.1.6 Phone status event

Provide two versions of the events have same functions for the phone. One does not contain caller ID, one that contains Caller ID information.Please select the appropriate event binding process according to their needs.

6.2 Short message

6.2.1 Send short message

Use the following code to send mail:

```
if(SMS.Sms.GetInstance().SMSSend(sDescAddress, sSMSContent))
{
//do things when send success
```

```
}
else
{
//do things when send fail
}
```

6.2.2 Event

1) Send success event

Binding events:

```
mSms.OnSMSSendSuccessEvent += new
SEUIC.Phone.SMS.Sms.SMSSendNotifyEvent(mSms_OnSMSSendSuccessEvent);
```

2) Failed to send short message event

Binding events:

```
mSms.OnSMSSendFailEvent += new
SEUIC.Phone.SMS.Sms.SMSSendNotifyEvent(mSms_OnSMSSendFailEvent);
```

3) Receive short message event

Binding events:

```
mSms.OnSMSReceivedEvent += new
SEUIC.Phone.SMS.Sms.SMSReceivedNotifyEvent(mSms_OnSMSReceivedEvent);
```

6.3 Dial-up

6.3.1 Dial

Use the following code to dial:

```
if (mRas.DialUp("", "", "*99***1#"))
{
//do things when ras connnect success
}
else
{
//do things when ras connect fail
}
```

6.3.2 Hang up

```
if (mRas. HangUp ("", "", "*99***1#"))
{
//do things when ras hangup success
}
else
{
//do things when ras hangup fail
}
```

6.3.3 Get connection status

Get the connection status of the enumerated type, the sample is as follows:

```
RAS.RASConnState rasConnState = mRas.GetStatus();
```

6.3.4 Event

Mainly to provide connectivity to successful events, and connection failure events. Need to bind according to their own logic.

6.4 Module

Module mainly provides control of the module, module status, access to information, access to network information and other methods. They are all the static type of method without instantiating the object.

6.4.1 Enable the module

```
if (Module.Module.ModuleEnable())
{
    //do things when ModuleEnable success
}
else
{
    //do things when ModuleEnable fail
}
```

6.4.2 Disable module

```
if (Module.Module.ModuleDisable())
{
//do things when ModuleDisable success
}
else
{
//do things when ModuleDisable fail
}
```

6.4.3 Reset module

```
if (Module.Module.ResetModule())
{
    //do things when ResetModule success

}
else
{
    //do things when ResetModule fail
}
```

6.4.4 Enable module anomaly detection

```
if (Module.Module. ModuleAbnormalDetectEnable())
{
//do things when ModuleAbnormalDetectEnable success
}
else
{
//do things when ModuleAbnormalDetectEnable fail
}
```

6.4.5 Disable the module anomaly detection

```
if (Module.Module. ModuleAbnormalDetectDisable())
{
    //do things when ModuleAbnormalDetectDisable success
}
else
{
    //do things when ModuleAbnormalDetectDisable fail
}
```

6.4.6 Get the module status

The module status returned is an enumerated type:

```
Module.ModuleStatus moduleStatus = Module.Module.GetModuleStatus();
```

6.4.7 Get the original information of the module name and version.

Return the original module:

```
string sModuleInfo = Module.Module.GetModuleInfo();
```

6.4.8 Get the type of wireless network

Return an enumerated type:

```
Module.NetworkType networkType= Module.Module.GetNetworkType();
```

6.4.9 Get electronic serial number ESN

Return the string type:

```
string sESN = Module.Module.GetESN();
```

This method is not yet achieved.

6.4.10 Get International Mobile Subscriber Identity IMSI

Return the IMSI of the string type:

```
string sIMSI = Module.Module.GetIMSI();
```

6.4.11 Get the mobile equipment identity number MEID

Return a string type, the MEID:

```
string sMEID = Module.Module.GetMEID();
```

6.4.12 Get the signal strength RSSI

0-7 of int type signal returned:

```
int iRSSI = Module.Module.GetRSSI();
```

Note: Returned signal values expressed in the range of 0-7

6.4.13 Get the network signal strength HDRRSSI of EVDO

Return the HDRRSSI which is int type :

```
int iRSSI = Module.Module.GetHDRRSSI();
```

6.4.14 Get the cell identity code CI

The return value is string type:

```
string sCellID = Module.Module.GetCellID();
```

6.4.15 Get the location area code LAC

The return value is string type:

```
string sLAC = Module.Module.GetLAC();
```

6.4.16 Send AT command directly to the module

The return value is string type:

```
string sATCMDResponse = Module.Module.SendATCommand(sATCMD);
```

6.4.17 Get the operation mode of EVDO

The return value is an enumeration type:

```
EVDOMode evdoMode= Module.EVDO.GetNetWorkMode();
```

6.4.18 Set the operation mode of EVDO

Return that if the setting is successful, its argument is an enumeration:

```
if(Module.EVDO.SetNetWorkMode())
{
//do things when set evdo mode success
}
else
{
//do things when set evdo mode fail
}
```

This method is not yet achieved

6.4.19 Get the location information of EVDO base station

The return value is EVDOLocationInfo class instance:

```
EVDOLocationInfo evdoLocationInfo= Module.EVDO.GetEVDOLocationInfo();
```

6.4.20 Get the Speaker Volume

Get the current speaker volume

Int iSpeakerVolume=GetSpeakerVolume();

6.4.21 Set the Speaker Volume

Set the current speaker volume. The parameter is the volume value. The return value is bool type, indicates whether the settings is successful.

```
bool SetSpeakerVolume(int iSpeakerVolume);
```

6.4.22 Get the microphone volume

Returns the current microphone volume

Int iSpeakerVolume= GetMicVolume();

6.4.23 Set the microphone volume

Set the current microphone volume. The parameter is the volume value. The return value is bool type, indicates whether the settings is successful.

```
bool SetMicVolume(int iSpeakerVolume);
```