Palm Handspring GSM/GPRS Phone Library Documentation

Author: Prashant Batra

Table of Contents

1.Overview

The Palm Phone Library is an interface to phone hardware. Its usage is increasingly gaining importance as developers are developing many applications involving the use of Phone APIs.

It is widely used in the organizations mainly responsible for developing application involving phone feature's usage, like Phone call recording, Black list (used to ignore a particular caller), caller identification, customized ringtones, conferencing, etc. One of the common software that shows the caller's image also uses the Phone Library. Not only this, it can be used to set the device on various modes like vibration, speaker, silent, etc.

The information about its usage is somehow very much limited. The lack of information makes the life of Palm developers challenging and even frustrating at times, causing loss of focus.

We faced this problem ourselves and thought to compile a document which can help our developer community to move to other more important matters rather than struggling with Phone API usage and implementation.

Goals

This document has been designed keeping following points in mind:

- Provide access to functionalities unique to client telephony/telephonic device.
- Assist developers in the rapid development of products involving the use of various phone features.

The Palm APIs discussed herein broaden the scope for development by providing a developer friendly and detailed descriptions.

How it is useful to ME?

Mostly used Phone API Features are discussed in this document.

You can use it to find the appropriate function, see its usage and implement it in your code quickly. Few of the features supported by Phone API are:

- Find the currently active line
- Maximum connections supported by the phone
- Find whether your cell phone module supports more than one connection at a time or not
- Find IMEI No
- Vibrate the Palm Device
- Find the number of active calls
- Set one of the various modes like speaker mode, headset mode, etc.
- Find a state of a phone call (active or not)

And, do much more…

For an exhaustive list of Phone API, please refer to the PhoneLib Ref provided by HandSpring.

2. Introduction

As we are getting more and more phone enabled Palm devices (Treo 600, 650, 700p etc.), the use of Phone API is gaining popularity. Somehow the information about its usage is limited and we had to go out of the way in-order to make our application's function as they do now.

This documentation is a compendium of our knowledge and provides details about the Phone Library included in Treo GPRS/GSM communicator family.

Detailed description with examples have been provided, wherever possible.

Note: The user may find differences in usage with Treo300, as it is a CDMA device.

3. Target Audience

This module is for Palm developers, beginners and experienced both, who wish to gain an understanding of Phone API and are responsible for building palm applications.

4. Notations

Throughout this document, we will refer to functions and type with **Bold Times New Roman**. Functions have their own subsections, and for each function we have provided the following:

Declared In
Header File in which the function is declared

Description
What the function actually does.

Parameters
Parameters used by the function.
Input parameter is indicated by symbol ->
Output parameter is indicated by symbol <-

Result
Return value from function call.

Example
An example to show the usage of the function.

The parameters of a function are referred using *Italics Times New Roman*.
Times New Roman font with Bold, Italic and Underline, represents the enum type
Lines with Bold and Italic option represent important Note.

5. API Functions: Descriptions and Examples

Function

Err PhnLibGetLineState(**UInt16** *refNum*, **UInt16** *line*, **PhnLineStatePtr** *lineState*)

Declared In

HsPhoneMisc.h

Description

This function retrieves the state of the phone line.
It basically gives you the version number of the firmware, active line, information related to divert indicator and the voice mail indicator.

Parameters

-> *refNum* Library reference number
-> *line* Line to query for
<- *lineState* State of the Line

Result

returns 0 for success; otherwise failed and returns an error code as defined in “PhoneLibrary Error Codes.”

Example

```
UInt16   pLib  =  sysInvalidRefNum;
UInt16   line;
PhnLineStatePtr   lineState;

line  =  0;
HsGetPhoneLibrary(&pLib);
PhnLibGetLineState(pLib, line, lineState);
```

Function

Err PhnLibGetRadioState(**UInt16** *refNum*, **PhnRadioStatePtr** *radioState*)

Declared In

HsPhoneMisc.h

Description

This function retrieves the state information about the radio.
It tells you the version number of the firmware, maximum connections supported by the phone, whether your cell phone module supports more than one connection at a time or not, and the line currently active.

Remember: if ALS (Active Line support) is not supported then currently active line should be 1.

Parameters

-> *refNum* Library reference number
<- *RadioState* State of the Radio

Result

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
UInt16  pLib = sysInvalidRefNum;
PhnRadioStateType pRadioState;

HsGetPhoneLibrary(&pLib);
PhnLibGetRadioState(pLib, &pRadioState);
```

Function

Boolean PhnLibCardInfo (**UInt16** *refNum*, **CharPtr** * *manufacturer*, **CharPtr** * *model*, **CharPtr** * *version*, **CharPtr** * *serial*)

Declared In

HsPhoneMisc.h

Description

This function retrieves the manufacturer, model, version of firmware and the serial number (IMEI) of your phone.

Parameters

->	*refNum*	Library reference number
<-	*manufacturer*	The Radio manufacturer Id. Set it to NULL if not needed
<-	*model*	The Radio model number. Set it to NULL if not needed
<-	*version*	Version of Radio Firmware. Set it to NULL if not needed
<-	*serial*	It is basically the IMEI number. Set it to NULL if not needed

Result

returns 0 for success.

Example

```
UInt16  pLib = sysInvalidRefNum;
CharPtr  serial;
CharPtr  model;
CharPtr  manufacturer;
CharPtr  version;

HsGetPhoneLibrary(&pLib);
PhnLibCardInfo (pLib, &manufacturer, &model , &version, &serial);
```

Function

Err PhnLibConnectionAvailable(**UInt16** *refNum*, **PhnConnectionEnum** *connection*, **Boolean** * *pAvailable*)

Declared In

HsPhoneMisc.h

Description

This function checks if the specified connection type is available for use or not.
The connection may be voice connection, CSD connection (Circuit Switched Data), GPRS connection or OneX connection..

Parameters

-> *refNum* Library reference number
-> *connection* Type of connection to check for.(See the PhnConnectionEnum)
<- *pAvailable* Result indicating if a connection is available

Result

returns 0 for success; otherwise failed and return an error code as defined in “Phone Library Error Codes.”

Example

Boolean pAvailable;
UInt16 pLib = sysInvalidRefNum;

HsGetPhoneLibrary(*&pLib*);
PhnLibConnectionAvailable(*pLib*, ***voiceConnection***, *&pAvailable*);

Function

Err PhnLibStartVibrate (**UInt16** *refNum*, **Boolean** *pulse*, **Boolean** *repeat*)

Declared In

HsPhoneMisc.h

Description

This function starts the vibrator motor and hence vibrates the phone.

Parameters

-> *refNum* Library reference number
-> *pulse* pulse motor
<- *repeat* Repeat sequence

Result

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Note: The function has been deplored. So you cant use this function for Vibration.

Instead, you can use **HsIndicatorState** (**UInt16** *count*, **UInt16** *indicatorType*, **UInt16** * *stateP*).

Example

UInt16 indicatorState = ***kIndicatorAlertAlert***;

HsIndicatorState(*2*, ***kIndicatorTypeVibrator***, *&indicatorState*);

Function

PhnLibGetCountByService(UInt16 *refNum,* **PhoneServiceClassType** *service*)

Declared In

HsPhoneMisc.h

Description

This function locates the call item that has the specified service and if such item exists, then returns the total number of call items that has the same service.
Actually there is a class of service for which applications can register:

```
enum _PhoneServiceClassType
  {
    phnServiceVoice = 1,

    phnServiceSMS = 2,

    phnServiceActivation = 4,

    phnServiceData = 8,

    phnServiceIOTA = 16,

    phnServiceSIMToolkit = 32,

    phnServiceAFLT = 64,

    phnServiceMisc = 128,

    phnServiceEssentials = 256,

    phnServiceMMS = 512,

    phnServiceWAP = 1024,

    phnServiceAll = phnServiceVoice | phnServiceSMS | phnServiceAFLT |
                    phnServiceData | phnServiceActivation | phnServiceMisc | phnServiceIOTA |
                    phnServiceEssentials | phnServiceSIMToolkit | phnServiceMMS | phnServiceWAP,

    phnServiceMax = 0xFFFF        // Reserved.
  };
typedef UInt16 PhoneServiceClassType;
```

Parameters

-> *refNum* Library reference number

-> *service* The specified connection service (can OR bit combination)

Result

returns number of call items that has the same status.

Example

UInt32 calls = 0;
UInt16 pLib = sysInvalidRefNum;

HsGetPhoneLibrary(*&pLib*);
calls = **PhnLibGetCallCountByService**(*pLib*, ***phnServiceVoice***);

'calls' will finally contains the number of calls active at that particular time.

So you can find the number of active calls by using the function.

Function

Err PhnLibGetEquipmentMode(**UInt16** *refNum*, **PhnEquipmentMode** * *equipmentMode*)

Declared In

HsPhoneMisc.h

Description

This function gives you the mode your cell is currently in. It basically determines the state the hardware is in.

For example: If your speaker is On during call, this function will return you the Speaker Phone Mode i.e. ***PhnSpeakerPhoneMode***.

Other modes may be Handset Mode, Headset Mode**,** CarKit Mode**,** HandsetLidClose Mode**,** Auto Mode (automatic mode where modem internally decides the mode), Bluetooth Handsfree Mode etc.

Parameters

-> *refNum* Library reference number
<- *equipmentMode* Phone Equipment Mode

retruns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
UInt16   pLib  =  sysInvalidRefNum;
PhnEquipmentMode  equipmentMode;

HsGetPhoneLibrary(&pLib);
PhnLibGetEquipmentMode(pLib, &equipmentMode);
```

Function

Err PhnLibSetEquipmentMode(**UInt16** *refNum*, **PhnEquipmentMode** *equipmentMode*)

Declared In

HsPhoneMisc.h

Description

This function sets equipment mode. The various modes that can be passed are as follows:

typedef enum
{
phnHandsetMode, //sets handset mode
phnHeadsetMode, //sets headdset mode
phnSpeakerPhoneMode, //sets speaker phone mode
phnCarKitMode, //set carkit mode
phnHandsetLidCloseMode,
phnAutoMode, //automatic mode where modem internaly decides the mode
phnBluetoothHeadsetMode,
phnBluetoothHandsfreeMode,
phnEquipmentModeLast

}PhnEquipmentMode;

The function is mostly used to switch between speaker phone and headset mode.

Parameters

-> *refNum* Library reference number
-> *equipmentMode* Phone Equipment Mode to set

Result
returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example
UInt16 pLib = sysInvalidRefNum;
HsGetPhoneLibrary(*&pLib*);
PhnLibGetEquipmentMode(*pLib*, ***phnSpeakerPhoneMode***);

This will set the Speaker Phone Mode

Function

PhnLibGetLibAPIVersion (**UInt16** *refNum*, **Uint32** **dwVerP*)

Declared In

HsPhoneMisc.h

Description

This function returns the version number as 32 bit unsigned integer.

Parameters

-> *refNum* Library Reference Number
<- *dwVerP* The version number. You need to allocate storage before calling it

Result

returns 0 for success: otherwise failed and return error code

Example

```
Uint16   pLib  =  sysInvalidRefNum;
Uint32 dwVerP;

HsGetPhoneLibrary(&pLib);
PhnLibGetLibAPIVersion(pLib, &dwVerP);
```

Function

PhnLibIsCallMuted (**UInt16** *refNum*, **Boolean** * *isMute*).

Declared In

HsPhoneMisc.h

Description

This function tells whether a phone is on mute or not.

Parameters

-> *refNum* Library reference number
<- *isMute*: parameter that will return 1 if phone call is mute

Results

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
Boolean   ismute;
UInt16    pLib  =  sysInvalidRefNum;

HsGetPhoneLibrary(&pLib);
PhnLibIsCallMuted (pLib, &ismute);
```

Function

PhnLibGetDeviceID (**UInt16** *refNum*, **CharPtr** * *deviceId*)

Declared In

HsPhoneMisc.h

Description

This function retrieves the device Id of the phone. The id is basically IMEI number

Parameters

-> *refNum* Library reference number
-> *deviceId* Id of device

returns 0 for success; otherwise failed and returns an error code as defined in "Phone Library Error Codes."

Example

```
CharPtr deviceId;
UInt16   pLib  =  sysInvalidRefNum;

HsGetPhoneLibrary(&pLib);
PhnLibGetDeviceID (pLib, &deviceId);
```

.

Function

Err PhnLibErrorRate (**UInt16** *refNum*, **WordPtr** *errorRate*).

Declared In

HsPhoneMisc.h

Description

This function returns the bit error rate (BER) of the wireless channel. The value will range from 0 to 7. A value of 99 indicates an unknown or non detectable signal.

Parameters

-> *refNum* Library reference number
<- *errorRate* Bit error rate

Result

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
WordPtr errorRate;
UInt16   pLib  =  sysInvalidRefNum;

HsGetPhoneLibrary(&pLib);
PhnLibErrorRate (pLib, errorRate);
```

Function

void PhnLibGetErrorText (**UInt16** refNum, **Err** error, **CharPtr** buffer, **UInt16** bufferLen).

Declared In

HsPhoneMisc.h

Description

This function translates the given error code and return a text string containing the error message.

Parameters

-> *refNum* Library reference number
-> *error* Error code
<- *buffer* String
-> *bufferLen* Size of string

Result

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
Err       error  =  phnErrNotOpen ;
UInt16   pLib  =  sysInvalidRefNum;
CharPtr  buffer;
UInt16   bufferLen;

HsGetPhoneLibrary(&pLib);
PhnLibGetErrorText (pLib, error,  buffer,  bufferLen);.
```

The example will return -- “the library not open “ error as string.

Function

PhnLibGetEchoCancellation(**UInt16** *refNum*, **Boolean** ***** *echoCancellationOn*).

Declared In

HsPhoneMisc.h

Description

This function is used to get the echo cancellation mode.

Parameters

-> *refNum* Library reference number
<- *echoCancellationOn* Echo cancellation mode

Results

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
UInt16   pLib  =  sysInvalidRefNum;
Boolean echoCancellationON;

HsGetPhoneLibrary(&pLib);
PhnLibGetEchoCancellation(pLib, &echoCancellationOn);
```

Function

Err PhnLibSetEchoCancellation(**UInt16** *refNum*, **Boolean** *echoCancellationOn*)

Declared In

HsPhoneMisc.h

Description

This function sets the echo cancellation mode.

Parameters

-> *refNum* Library reference number
<- *echoCancellationOn* Echo cancellation mode to set

Results

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

UInt16 pLib = sysInvalidRefNum;
Boolean echoCancellationON = 0;

HsGetPhoneLibrary(*&pLib*);
PhnLibSetEchoCancellation(*pLib*, *&echoCancellationOn*);

Function

PhnLibGetMicrophone (**UInt16** *refNum*, **Int16 *** *gain*).

Declared In

HsPhoneMisc.h

Description

This function is used to get the microphone gain.

Parameters

-> refNum Library reference number
<- gain Microphone gain

Results

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
UInt16   pLib  =  sysInvalidRefNum;
Int16    gain;

HsGetPhoneLibrary(&pLib);
PhnLibGetMicrophone (pLib, &gain);
```

Function

PhnLibSetMicrophone (**UInt16** *refNum*, **Int16** *gain*).

Declared In

HsPhoneMisc.h

Description

This function sets the microphone gain.

Parameters

-> *refNum* Library reference number
-> *gain* Microphone gain

Results

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

```
UInt16   pLib  =  sysInvalidRefNum;
Int16    gain  =  10;

HsGetPhoneLibrary(&pLib);
PhnLibSetMicrophone (pLib, gain);
```

Function

PhnLibSleep(**UInt16** *refNum*).

Declared In

HsPhoneLibrary.h

Description

This function puts the phone library to sleep.

Parameters

-> *refNum* Library reference number

Results

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

UInt16 pLib = sysInvalidRefNum;

HsGetPhoneLibrary(*&pLib*);
PhnLibSleep(*pLib*);

Function

PhnLibWake(**UInt16** *refNum*).

Declared In

HsPhoneLibrary.h

Description

This function awakens the phone library.

Parameters

-> *refNum* Library reference number

Results

returns 0 for success; otherwise failed and returns an error code as defined in “Phone Library Error Codes.”

Example

UInt16 pLib = sysInvalidRefNum;

HsGetPhoneLibrary(*&pLib*);
PhnLibWake(*pLib*);