Phone Library Application Development on the Treo 300

Introduction

The Treo 300 is a CDMA version of the Treo GSM products: Treo 180, 180g, 270. Although work was done to ensure that the Phone Library API (include SMS API) would be identical, there are some minor differences when running on the Treo 300. Although we cannot guarantee this, we are working to resolve these differences in future updates to the Treo 300.

Phone Library Name

The name of the Phone Library on the Treo 300 is different than the library name on the Treo 180/270. The following table outlines the differences:

GSM Product	CDMA Product
Phone Library Name `GSMLibrary.lib`	*Phone Library Name* `PhoneLib.prc`
Phone Library Database Type `libr`	*Phone Library Database Type* `libr`
Phone Library Creator ID `GSM!`	*Phone Library Creator ID* `HsCL`

When opening the library, one must use the appropriate library name and creator ID to the `SysLibFind` and `SysLibLoad` calls.

Detecting the Device Type

See the "Handspring GSM/GPRS Phone Library Reference" document for more details.

Message Database

See the "Handspring GSM/GPRS Phone Library Reference" document for more details.

Launch Codes

The Phone Library on both products can send different launch codes to registered applications based on what the system is doing. The launch code that is used by the library to indicate a Phone Library event is, `phnLibLaunchCmdEvent`. On the GSM products this is defined to be `0x2bad`. On the CDMA products, this is defined to be `0x7000`. Applications should respond to both launch codes to ensure that they can receive the code on both products.

Registration

The GSM Product will send out a `phnLibLaunchCmdRegister` to applications to let them know they can register with the system. The CDMA product does not send out such event. Developers can use the reset launch command (`sysAppLaunchCmdSystemReset`) from the system to register with the Phone Library.

Phone Event Code

The Phone Event Codes (enum) is a bit different on the GSM Product than on the CDMA product. Based on the product you are running on, you may have to use the different enums. The list below shows the different enums.

CDMA Product Phone Event Code

```
// Phone event types
typedef enum
  {
    phnEvtCardInsertion,      // PhoneLib is installed!
    phnEvtRegistration,       // Phone able to find service!
    phnEvtError,              // indicator of something important happens to
                        // the phone that needs to bring up alert
    phnEvtKeyPress,           // Phone, data or power button pressed.
    phnEvtPower,              // Phone is at the end of Power up sequence
                                 // or phone starts the power down process
                                 // or is at the end of power off sequence
    phnEvtProgress,           // indicate that an outgoing call in
                                 // dialing state needs to be created
    phnEvtIndication,         // network search banner or power save
                                 // banner needs to be drawn
    phnEvtConnectInd,         // indicate that an incoming call in
                                 // incoming state needs to be created
    phnEvtConnectConf,        // The call in dialing or incoming state
                                 // with specified ID is just connected
    phnEvtSubscriber,         // Need to update the number and the name
                                 // of the specific connection ID
    phnEvtDisconnectInd,      // A specific connection ID is told to
                                 // shut down suddenly
    phnEvtDisconnectConf,     // An ACK for a disconnection command on a
                                 // specific connection ID is received
    phnEvtBusy,               // Network Busy condition is received
    phnEvtUpdate,             // PhoneUI and registered application need
                                 // to update its view.
    phnEvtConference,         // Modem is in 3-way call mode now.
    //SMS support
    phnEvtVoiceMail,          // Voicemail indicator has been received
    phnEvtMessageInd,         // A new SMS message has just been
                                 // received (CMT in IS-637 standard)
    phnEvtWAPInd,             // A new SMS message has just been
                                 // received (WAP in IS-637 standard)
    phnEvtSegmentInd,
    phnEvtMessageStat,
    phnEvtMessageDel,
    phnEvtMessageMoved,
    phnEvtMemoryFull,         // Memory is full, cannot accept any more messages
    phnEvtMemoryOK,           // Memory is not full anymore,can receive messages
    // filler
    phnEvtFiller1,
    phnEvtFiller2,
    phnEvtFiller3,

    // CR 1608: Phone debug screen update needs to be a separate event
    phnEvtDebugReport,        // A field in the phone debug report has
                                  // changed the status.
    // End of CR 1608
    phnEvtSSModeChanged,      // Phone just change to new System Selection Mode
    phnEvtForceUpdateTitleBar,  // Title bar needs to be updated!
    //CR-1337
    phnEvtFlipSliderSwitch,   // Slider switch just change the position, need
                        // to call PhnLib to have the exact position
    phnEvtPDDataChanged,         // phnEvtPDDataChanged,
    phnEvtOneXStatus,            //indaictes 1X avail state, packet data
                              // avail state, MIP avail state -
    phnEvtMIPFailed,       //indicates mobile IP failed
    phnEvtPhoneEquipmentMode, // change in equipment mode
    phnEvtIOTAStatus,            //indicates IOTA status change
```

```
    kMaxPhnEvtSupported        // Must be at last position.
  }
PhnEventCode;
```