Reviewer Approved

Simulation of keys can be used in numerous Applications where we need to programmatically simulate key events for various purposes.

Simulating key events can be done in the following ways:

Contents

- 1 Procedure1: Using TKeyEvent:
- 2 Procedure2: Using TWsEvent:
- 3 Procedure3: Using TApaTask:
- 4 Procedure4: Using TRawEvent:
- 5 Procedure5: Using SimulateRawEvent:
- 6 Special case

Procedure1: Using TKeyEvent:

First set the iCode,iScanCode,iModifiers & iRepeats for the key to be simulated of the TKeyEvent class. Using RWsSession call the SimulateKeyEvent() by passing the keyevent to be generated.

The following is the sample code for simulation of green key:

```
Location: W32STD.H// RWsSession
Link against: ws32.lib  // RWsSession

Location: W32STD.H// TKeyEvent

        RWsSession wsSession = CCoeEnv::Static()->WsSession();
        TKeyEvent keyEvent
        keyEvent.iCode = EKeyYes;  //member of TKeyCode
        keyEvent.iScanCode = EStdKeyYes;
        keyEvent.iModifiers = 0;
        keyEvent.iRepeats = 0;
        wsSession.SimulateKeyEvent(keyEvent);
        wsSession.Flush();
```

Note that at least in Symbian OS v9.3, this method is protected with the capability: **SwEvent**.

NOTE: This seems to be working in S60 V3 MR emulator without capabilities, but produces WSERV 66 panic, when run on a device without capabilities.

NOTE2: This method does exists in S60 V3 MR SDK headers, but not in the documentation.

Procedure2: Using TWsEvent:

Simulation of a key can also be accomplished by using TWsEvent class, wherein the event key to be simulated is passed to the RWsSession by calling the SendEventToWindowGroup() and also passing the id of the focus window. This API sends an event to a window group.

The following is the sample code for simulation of Up Arrow key:

```
Location: W32STD.H// RWsSession
Link against: ws32.lib  // RWsSession

Location: W32STD.H// TWsEvent

        TWsEvent  event;
        RWsSession wsSession = CCoeEnv::Static()->WsSession();
        TInt id = wsSession.GetFocusWindowGroup();
      SetType(EEventKey);
      SetTimeNow();
      KeyEvent.iCode = EKeyUpArrow;
      KeyEvent.iModifiers = 0;
      KeyEvent.iRepeats = 0;
      KeyEvent.iScanCode = EStdKeyUpArrow;
        wsSession.SendEventToWindowGroup( id, event );
        wsSession.Flush();
```

or

```
Location: W32STD.H// RWsSession
Link against: ws32.lib  // RWsSession

Location: W32STD.H// TWsEvent

        TWsEvent  event;
        RWsSession wsSession = CCoeEnv::Static()->WsSession();
        TInt id = wsSession.GetFocusWindowGroup();
      SetType(EEventKey);
      SetTimeNow();
        event.Key()->iCode = EKeyUpArrow;
      KeyEvent.iModifiers = 0;
      KeyEvent.iRepeats = 0;
      KeyEvent.iScanCode = 0;
        wsSession.SendEventToWindowGroup( id, event );
```

The output of both codes are same.

Note that at least in Symbian OS v9.3, this method is protected with the capability: **SwEvent**.

Procedure3: Using TApaTask:

This API finds the app using TApaTaskList, and sends the required Key event to that App.

```
Location: apgtask.h  // TApaTaskList
Link against: apgrfx.lib  // TApaTaskList

Location: apgtask.h  // TApaTask
Link against: apgrfx.lib  // TApaTask

TApaTaskList tlist(iEikonEnv->WsSession());
TApaTask app(tlist.FindApp(_L("AppName")));

TKeyEvent key;
key.iModifiers = 0;
```

```
key.iRepeats = 0;
key.iCode = EKeyDownArrow;
key.iScanCode = EStdKeyDownArrow;

app.SendKey(key);
```

Note that at least in Symbian OS v9.3, this method is protected with the capability: **SwEvent**.

Procedure4: Using TRawEvent:

Simulation can also be done using the TRawEvent class by Set() API passing the respective keydown/keyup event and the scancode of the key.After setting the event, add this event to the usersrv.

The following is the sample code for simulation of Red key:

```
Location: e32event.h       // TRawEvent
Link against: euser.lib    // TRawEvent

Location: e32std.h         // User class
Link against: euser.lib    // User class

        TRawEvent lEventDown;
        lEventDown.Set(TRawEvent::EKeyDown, EStdKeyNo);
       UserSvr::AddEvent(lEventDown);
    User::After(100000);
        TRawEvent lEventUp;
        lEventUp.Set(TRawEvent::EKeyUp, EStdKeyNo);
       UserSvr::AddEvent(lEventUp);
```

Note that in Symbian OS v9.x, this method is protected with the capability: **SwEvent**.

Procedure5: Using SimulateRawEvent:

```
RWsSession::SimulateRawEvent(TRawEvent aEvent)
```

This function should only be used for test code. Note that in Symbian OS v9.x, this method is protected with the capability: **SwEvent**.

Special case

If you do not wish to simulate a key event, but open a dialog, with keys 0-9 or so, so that the opening key will appear in the dialog, you can just open the dialog with EEventKeyDown, so the EEventKey will be directed to the dialog itself. This does not require capabilities.