How to capture Keyevents in thread or exe

The code below shows how a thread can capture some keys and react to them, optionally sending them to the top app if it's not interested in them itself. You'll only need a cleanup stack active for this thread, no active scheduler needed.

Headers Required:

```
#include <W32STD.H>
#include <APGWGNAM.H>
#include <e32keys.h> //for key codes
```

Libraries Required:

```
LIBRARY ws32.lib
LIBRARY apgrfx.lib
```

Source:

```
      // connect to window server
 RWsSession ws;
 User::LeaveIfError(ws.Connect());
 CleanupClosePushL(ws);

 TRequestStatus status;

 // create a window group for the thread
 RWindowGroup wg(ws);
 wg.Construct((TUint32)&wg, EFalse);
 CleanupClosePushL(wg);

 // capture a key
 User::LeaveIfError(wg.CaptureKey(EKeyLeftArrow, 0, 0));

 // listen for the key presses
 ws.EventReady(&status);

 // hide this window group from the app switcher
 wg.SetOrdinalPosition(-1);
 wg.EnableReceiptOfFocus(EFalse);
 CApaWindowGroupName* wn=CApaWindowGroupName::NewLC(ws);
 wn->SetHidden(ETrue);
 wn->SetWindowGroupName(wg);

 // handle key events
 for(;;) {
  User::WaitForAnyRequest();
  if (status.Int()==KErrNone) {
   TWsEvent e;
   ws.GetEvent(e);

   TInt c;
   TKeyEvent* aKeyEvent=e.Key();
   c=aKeyEvent->iCode;
```


(C) Copyright Nokia 2014. All rights reserved.

```
    // do something with keypress

    // if not ours, then send to top window group
    // note that this breaks key repeat
            TInt wgid=ws.GetFocusWindowGroup();
    ws.SendEventToWindowGroup(wgid, e);
   }
   ws.EventReady(&status);

   // stop condition
  }

  // clean up
  ws.EventReadyCancel();
  CleanupStack::PopAndDestroy(3); //ws, wg, wn
```

Related Links:

- How to get ReDraw event in exe from window server?
- How to use graphics in a non-GUI Symbian exe
- Archived:How to launch an Exe and pass command line arguments
- How to start EXE from an EXE in 3rd Edition
- Archived:Displaying UI controls in Symbian console exe programs

(C) Copyright Nokia 2014. All rights reserved.