

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
::LeaveIfError(ws.Connect());
CleanupClosePushL

TRequestStatus status

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

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

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

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

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

        ; TInt c
        * aKeyEvent = e.KeyEvent
        = aKeyEvent->iCode; c

// do something with keypress

// if not ours, then send to top window group
// note that this breaks key repeat
=ELCGetTopWindowGroup();
SendEventToWindowGroup(wgid, e);
}
```

How_to_capture_Keyevents_in_thread_or_exe

```
EventReady(&swatus);  
  
// stop condition  
}  
  
// clean up  
EventReadyCancel();  
CleanupSpAndDestroy(3); //ws, wg, wn
```

Related Links:

- [How to get ReDraw event in exe from window server?](#)
- [Graphics in EXE](#)
- [How to Launch an EXE and Pass Command Line Arguments](#)
- [How to start EXE from an EXE in 3rd Edition](#)
- [Displaying controls in Symbian exe programs](#)