



The **CKeyCapturer** implementation illustrates how to capture key events in a background process. Note that you need to define the key to be captured when using the *CaptureKey()* function (example is capturing the **EKeyDevice0** key).

Also note that if the key should also be usable for the focused application, you need to manually forward it. This is already handled by the example as shown in the *RunL()* method. The callback interface function will return *EFalse* if the key is supposed to be forwarded to the focus application and *ETrue* if the key was consumed.

This example code should work with all platforms, but when used with S60 3rd Edition devices you need to have the SwEvent capability.

If you want to capture basic application keys in a foreground application you can use the standard *OfferKeyEventL()* method implemented in your **CCoeControl** derived class. If your application does not implement the standard application framework, you could also try using CKeyCapturer2 implementation.

### Note!

This will not capture volume keys or the multimedia keys such as the play key on the headset.

For this you will need to use TSS000432 - Utilising media keys

### Header required:

```
#include <w32std.h>
#include <apgwnam.h> //CApaWindowGroupName
```

### Library Required:

```
LIBRARY ws32.lib
LIBRARY apgrfx.lib //CApaWindowGroupName
```

### Capability Required:

```
capability SwEvent
```

## CapturingKeys.cpp

```
CKeyCapturer* CKeyCapturer::NewL(MKeyCallBack& aObserver)
{
    CKeyCapturer* CKeyCapturer::NewLC(aObserver);
    CleanupStack(self);
    return self;
}

CKeyCapturer* CKeyCapturer::NewLC(MKeyCallBack& aObserver)
{
    CKeyCapturer* new (ELeave) CKeyCapturer(aObserver);
    CleanupStack(self);
    ->ConstructL();
}
```

## Capturing\_keys\_in\_background

```
return self;
}

CKeyCapturer::CKeyCapturer(MKeyCallBack& aObserver)
: CActive(EPriorityStandard), iObserver(aObserver), iHandle(-1)
{
}

CKeyCapturer::~CKeyCapturer()
{
if(iHandle > -1)
{
    CancelCaptureKey(iHandle);
}

    #Handle

    ()Cancel

    CloseWg;
    iWsSession();
}

void CKeyCapturer::ConstructL()
{
    CActiveScheduler::Add(this);

    ::LeaveIfError(iWsSession.Connect());

    =RWindowGroup(iWsSession);
    ::LeaveIfError(iWg.Construct((TUint32)&iWg, EFalse));
    SetOriginalPosition(-1);
    EnableReceiptOfFocus(EFalse);

    CApaWindowGroupName::NewLC(iWsSession);
->SetHidden(ETrue);
->SetWindowGroupName(iWg);
    CleanupStackAndDestroy();

    #HandleCaptureKey(EKeyDevice0, 0, 0);
    ()Listen
}

void CKeyCapturer::RunL()
{
if (iStatus == KErrNone)
{
    ; TWSEvent e
    GetEventFromSession.

if(iObserver.KeyCapturedL(e))
{
    = iWsSession.GetFocusWindowGroup();
    SendEventToWindowGroup(wgId, e);
}
}

if (iStatus != KErrCancel)
{
    (); Listen
}
}
```

## Capturing\_keys\_in\_background

```
void CKeyCapturer::DoCancel()
{
    iWsEventReadyCancel();
}

void CKeyCapturer::Listen()
{
    iWsEventReady(&iStatus);
    SetActive
}
}
```

## CapturingKeys.h

```
class MKeyCallBack
{
public:
    virtual TBool KeyCapturedL(TWsEvent aEvent) = 0;
};

class CKeyCapturer : public CActive
{
public:
    static CKeyCapturer* NewL(MKeyCallBack& aObserver);
    static CKeyCapturer* NewLC(MKeyCallBack& aObserver);
    virtual ~CKeyCapturer();
private:
    CKeyCapturer(MKeyCallBack& aObserver);
    void ConstructL();
    void RunL();
    void DoCancel();
    void Listen();
private:
    MKeyCallBack& aObserver
    RWsSession      iWsSession ;
    RWindowGroup    iWg
    TInt            ;          iHandle
};
```

If you need to capture keys that are intended for applications like Google Maps or Nokia Maps, where a key held down makes multiple events you might better use this code in the RunL function:

```
void CKeyCapturer::RunL()
{
    if (iStatus == KErrNone)
    {
        TWsEvent e;
        iWsSession.GetEvent(e);

        //Only if iRepeats is 0 we should forward the event
        if(e.Key()->iRepeats == 0 && iObserver.KeyCapturedL(e))
        {
            TInt wgId = iWsSession.GetFocusWindowGroup();
            iWsSession.SendEventToWindowGroup(wgId, e);
        }
    }

    if (iStatus != KErrCancel)
    {

```

## Capturing\_keys\_in\_background

```
        Listen();  
    }  
}
```

### Related Links:

- [Capturing all keys in Non-GUI applications](#)