Bring application to foreground with a keypress

Inorder to capture the keys while you application under background you've to override CCoeAppUi::HandleWsEventL to your AppUi. With the use of RWindowGroup::CaptureKeyUpAndDowns() you can capture the keys from an application. Below is some sample code which will show you basically how to do it, you may found some additional bit of code which you may use.

The mmp file

```
CAPABILITY SwEvent
```

The header file

```
/**
 * To store the handle to the captured key.
 */
TInt iHashKeyHandle;
```

The cpp file

```
// ----------------------------------------------------------------------------
// CMyAppUi::ConstructL()
// Symbian 2nd phase constructor can leave.
// ----------------------------------------------------------------------------
//
void CMyAppUi::ConstructL()
    {
    BaseConstructL();

    // ...

    // set application as system application so that it will
    // not be closed by system events
    CEikonEnv::Static()->SetSystem( ETrue );

    // capture hash '#' key permanently
    iHashKeyHandle = CEikonEnv::Static()->RootWin().CaptureKeyUpAndDowns(EStdKeyHash, 0,
0);

    // set application priority to foreground priority even if it goes to background
    CEikonEnv::Static()->WsSession().ComputeMode( RWsSession::EPriorityControlDisabled
);

    // Make the application a high priority application
    CEikonEnv::Static()->RootWin().EnableReceiptOfFocus( ETrue );
    CEikonEnv::Static()->RootWin().SetOrdinalPosition(0, ECoeWinPriorityAlwaysAtFront);
    }

// ----------------------------------------------------------------------------
// CMyAppUi::~CMyAppUi()
```


(C) Copyright Nokia 2014. All rights reserved.

```
// Destructor.
// ---------------------------------------------------------------------------
//
CMyAppUi::~CMyAppUi()
    {
    // Release the hanlde to the captured key.
    CEikonEnv::Static()->RootWin().CancelCaptureKeyUpAndDowns(iHashKeyHandle);

    // ...
    }

// ---------------------------------------------------------------------------
// CMyAppUi::HandleForegroundEventL(TBool aForeground)
// Handles changes in keyboard focus when the application is brought to the
// foreground, or put into the background.
// ---------------------------------------------------------------------------
//
void CMyAppUi::HandleForegroundEventL(TBool aForeground)
    {
    // Make the application a high priority application
    if (aForeground)
        {
        CEikonEnv::Static()->RootWin().SetOrdinalPosition(0,
ECoeWinPriorityAlwaysAtFront);
        }
    }

// ---------------------------------------------------------------------------
// CMyAppUi::HandleWsEventL (const TWsEvent &aEvent, CCoeControl *aDestination)
// Handles events sent to the application by the window server.
// ---------------------------------------------------------------------------
//
void CMyAppUi::HandleWsEventL (const TWsEvent &aEvent, CCoeControl *aDestination)
    {
    if (aEvent.Type() == EEventKey ||
        aEvent.Type() == EEventKeyUp ||
        aEvent.Type() == EEventKeyDown ||
        aEvent.Type() == EEventKeyRepeat)
        {
        // This is for switching back to Application when Hash key is pressed
        if (EStdKeyHash == aEvent.Key()->iScanCode && EEventKey == aEvent.Type())
            {
            TApaTask task(iEikonEnv->WsSession());
            task.SetWgId(CEikonEnv::Static()->RootWin().Identifier());
            task.BringToForeground();

            return;
            }
        }

        CAknAppUi::HandleWsEventL(aEvent, aDestination);
    }
```


(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.