



å| ä½ ä, ç "ç" å° æj æ ¶æ ¥ç» å ¶å± å¹

```
iWindowGroup *p;  
iWsSession *p;  
}  
  
void CWindowDrawer::ConstructL(void)  
{  
    CActiveScheduler *p;  
  
    LeaveIfError(iWsSession.Connect());  
  
    iScreenDevice *p;  
    LeaveIfError(iScreenDevice->Construct());  
    LeaveIfError(iScreenDevice->CreateContext(iWindowGc));  
  
    TFontSpec MyFont(KESpeArial, 12*15);  
    MyFont.Typeface.SetIsProportional(ETrue);  
    LeaveIfError(iScreenDevice->GetNearestFontInTwips(iMyFont, MyFontSpec));  
  
    iWindowGroup *p;  
    LeaveIfError(iWindowGroup.Construct((TUint32)&iWindowGroup, EFalse));  
    // iWindowGroup.SetOrdinalPosition(0, ECoeWinPriorityNormal);  
    iWindowGroup.SetOrdinalPosition(0, ECoeWinPriorityAlwaysAtFront);  
    iWindowGroup.SetReceiptOfFocus(EFalse);  
  
    CApaWindowGroup *p;  
    winGroup.SetHidden(ETrue);  
    winGroup.SetWindowGroupName(iWindowGroup);  
    CleanupAndDestroy();  
  
    =RWindow(iWsSession);  
    LeaveIfError(iWindow.Construct(iWindowGroup, (TUint32)&iWindow));  
  
    TPixelsTwipsAndRotation SizeAndRotation;  
    iScreenDevice->GetDefaultScreenSizeAndRotation(SizeAndRotation);  
  
    AWindow *p;  
    SetEvent(TPoint(0,0), SizeAndRotation.iPixelSize);  
    SetBackgroundColor(KRgbBlue);  
    // iWindow.SetOrdinalPosition(0, ECoeWinPriorityNormal);  
    SetOrdinalPosition(0, ECoeWinPriorityAlwaysAtFront);  
    SetWindowing(ETrue);  
    SetResizable(ETrue);  
  
    (); Draw  
    iWsRedrawReady(&iStatus);  
    SetActive  
}  
  
void CWindowDrawer::RunL()  
{  
    if (iStatus != KErrCancel)  
    {  
        TWsRedrawEvent e  
        GetRedrawEventFromSession.  
  
        // if Windows Server does not want a redraw the window handle is 0  
        if (e.Handle() != 0)  
        {  
            // draw our only window  
            (); Draw  
        }  
    }  
}
```

å| ä½ ä, ç "ç" å° æj æ ¶æ ¥ç» å ¶å± å¹

```
        RedrawWindow(iWindow, iRect, iOldStatus);
    };    SetActive
}
}

void CWindowDrawer::Draw(void)
{
    iWindow->Activate(iWindow);

    TRect(DrRect(0,0), iWindow.Size());

    iWindow->Update(DrwRect);
    iWindow->BeginDraw();
    iWindow->EndDraw(DrwRect);

    if(iMyFont)
    {
        ->UseFont(iMyFont);

        ((DrRect.Height() - iMyFont->HeightInPixels()) / 2);

        ->DrawText(iWindow, iHelloThere, TRect(0, StartY, DrwRect.Width(), (StartY + iMyFont-
>HeightInPixels())), iMyFont->AscentInPixels(), CGraphicsContext::ECenter, 0);
    }

    iWindow->Redraw();

    iWindow->Deactivate();
    iWindow->Flush();
}

void CWindowDrawer::DoCancel()
{
    iWindow->ReadyCancel();
}
}
```

## WindowDrawer.h

```
#include <gdi.h> // CFont
#include <w32std.h> //RWindowGroup, CWScreenDevice, CWindowGc

_LIT(KtxHelloThere "Hi there !");
_LIT(KFontArial "Arial");

class CWindowDrawer : public CActive
{
public:
    static CWindowDrawer* NewL(void);
    static CWindowDrawer* NewLC(void);
    ~CWindowDrawer() {}
protected:
    void DoCancel();
    void RunL();
private:
    CWindowDrawer() {}
    void ConstructL(void);
    void Draw(void);
private:
    RWSession ;
    iWSession
```

å| ä½ ä, ç "ç" å° æj æ ¶æ ¥ç» å ¶å± å¹

```
CWindowGc ; iWindowGc
CWScreenDevice iScreenDevice
RWindowGroup ; iWindowGroup
RWindow ; iWindow
* CFont; iMyFont
};
```

**????:**

- How to get ReDraw event in exe from window server?
- How to capture Keyevents in thread or 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
- Graphics in EXE

????: