

# Archived:Rendering on top of the video

 Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template `{{ReviewForRemovalFromArchive|user=~~~~~|write your reason here}}`.

## Purpose

This article shows how to render content on top of the video besides playing the video.

## Use cases

Scrolling some text content on top of the video besides playing the video.

Describing the content of the video.

## Description

There are no specific API's available for rendering content on top of the video. The only possibility is to go for a work around using `CDirectScreenAccess` and `CVideoPlayerUtility` classes.

However the use of `CDirectScreenAccess` class for rendering content on top of the video is working only for 3.0 devices and is not applicable for 3rd Edition FP1 devices as the content goes behind the video.

Also note that it causes flickering effect which is quite unacceptable. Hence the use of `CDirectScreenAccess` is just a workaround and not an appropriate solution for rendering on top of video as it works fine only on specific devices.

## Code Snippet

The rendering part can be done using the `CDirectScreenAccess` class.

Intially go for creating an object of type `CDirectScreenAccess` and add it to the `ActiveScheduler`, followed by a call to `StartL()` function. The `StartL()` function informs the window server that you are going to start direct screen access and sets up a graphics context with which you can draw to the screen.

```
iDSA = CDirectScreenAccess::NewL(iWSession,iWScreenDevice,iWindow,*this);
CActiveScheduler::Add(this);
StartL();
```

In the `StartL()` we can include the functionality for getting the graphics context, the region the DSA can draw in and finally setting the display to the defined region, using the following lines of code.

```
iGc = iDSA->Gc(); //iGc is an object of type CFbsBitGc
iGc->SetBrushStyle(CGraphicsContext::ENullBrush);
iRegion = iDSA->DrawingRegion();
iGc->SetClippingRegion(iRegion);
```

Using various API's available in `CFbsBitGc` class, the content can actually be drawn to the screen in the required format. More information is provided in the link below where you can get a samle example for rendering on top of the video.

## Known Issues

With respect to 3rd Edition FP1 devices, it's not possible to render on top of the video, but we can have an alternative of setting the video to some part of the screen and utilize the remaining portion for displaying the content to be rendered. Resizing the window in which the video is being played can be done using `SetDisplayWindowL()`.

## Example application

---

Sample application: [File:RenderingText.zip](#)