Archived:Accessing display memory with UserSvr::ScreenInfo() is deprecated from S60 3rd Edition onwards (Known Issue)

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*}}.

`UserSvr::ScreenInfo()` provides an address to the LCD display frame buffer. However, using this method of drawing directly to the screen does not work properly on S60 3rd Edition devices, and is considered as deprecated.

Description

The address `TScreenInfoV01::iScreenAddress`, obtained by calling the `UserSvr::ScreenInfo()` function, can be used for drawing directly to the frame buffer of a display. After modifying the frame buffer, the display still needs to be forced to update itself.

Prior to S60 3rd Edition, this could be done by generating a redraw event:

```
TRawEvent redraw;
redraw.Set( TRawEvent::ERedraw );
UserSvr::AddEvent( redraw );
```

On S60 3rd Edition devices this does not have any immediate effect; the screen is updated only after notifying the screen device about the out of date region(s).

Solution

Instead of retrieving the frame buffer address with `UserSvr::ScreenInfo()`, developers should start using `CDirectScreenBitmap` class for direct screen access. For more information, see the document S60 Platform: Scalable Screen-Drawing How-To.

As a workaround solution, it is still possible to use the old drawing method and force the screen to update itself by specifying the out of date region with

```
void CFbsScreenDevice::Update( const TRegion &aRegion );
```