



<b>ID</b>		<b>Creation date</b>	November 26, 2008
<b>Platform</b>	S60 3rd Edition, FP2	<b>Tested on devices</b>	S60 3rd Ed. FP2 SDK
<b>Category</b>	Java	<b>Subcategory</b>	eSWT, Splash Screen

**Keywords (APIs, classes, methods, functions):** eSWT, Splash Screen, Shell, FormLayout

## Overview

This snippet shows how to create a splash screen using eSWT. Example implements the splash screen as a window that is not maximized. Shows an image and also updates a ProgressBar. Follow the video link below to see how it works

[Splash Screen Example Video](#)

## Source file

```
import javax.microedition.midlet.MIDlet;
import org.eclipse.swt.SWT;
import org.eclipse.swt.events.SelectionEvent;
import org.eclipse.swt.events.SelectionListener;
import org.eclipse.swt.graphics.GC;
import org.eclipse.swt.graphics.Image;
import org.eclipse.swt.graphics.Rectangle;
import org.eclipse.swt.layout.FormAttachment;
import org.eclipse.swt.layout.FormData;
import org.eclipse.swt.layout.FormLayout;
import org.eclipse.swt.widgets.Display;
import org.eclipse.swt.widgets.Label;
import org.eclipse.swt.widgets.ProgressBar;
import org.eclipse.swt.widgets.Shell;

public class SplashScreenDemo extends MIDlet implements Runnable, SelectionListener {

    // A handle to the eSWT UI thread created by this MIDlet.
    private Thread UIThread;
    // The eSWT Display created by this MIDlet in the eSWT UI thread.
    // When this is created the MIDlet gets connected to the native UI
    // functionality and eSWT UI toolkit is initialised for it.
```

## How\_to\_create\_a\_Splash\_Screen\_in\_eSWT

```
private Display display;
// A Shell widget created by this MIDlet.
private Shell shell;
// A boolean to set when the event loop should exit.
private boolean exiting = false;

public void startApp() {
    // Create the eSWT UI thread.
    if(UIThread == null) {
        UIThread = new Thread(this);
        UIThread.start();
    }
}

public void pauseApp() {
    // Here we could reduce the resources but we should keep the Display
    // instance and the eSWT UI Thread.
}

// destroyApp is called when the MIDlet is terminated from the task list
// with the clear key, or the end key is pressed when the MIDlet is focused.
// It might also be called when the system needs to close applications
// e.g. in low memory conditions.
public void destroyApp(boolean unconditional) {
    // Make the event loop exit in the eSWT UI thread.
    exitEventLoop();
    // Wait for the eSWT UI thread to die.
    try {
        UIThread.join();
    } catch(InterruptedException e) {
    }
}

// This method can be called from any thread to make the event loop to exit.
void exitEventLoop() {
    exiting = true;
    Display.getDefault().wake();
}

// The eSWT UI Thread.
public void run() {
    // Create the Display.
    display = new Display();

    shell = new Shell(display);

    Rectangle screenSize = display.getBounds();
    final Image image = new Image(display, screenSize.width-50, screenSize.height/2);
    = GC.newGC(image);
    setBackground(display.getSystemColor(SWT.COLOR_CYAN));
    fillRectangle(image.getBounds());
    drawText("Splash Screen", 0, 5);
    dispose();
    final Shell splash = new Shell(shell, SWT.ON_TOP);
    final ProgressBar bar = new ProgressBar(splash, SWT.SMOOTH);
    Label label = new Label(splash, SWT.BORDER);
    setImage(image);
    FormLayout layout = new FormLayout();
    splash.setLayout(layout);
    FormData labelData = new FormData();
    labelData.new FormAttachment(100, 0);
```

## How\_to\_create\_a\_Splash\_Screen\_in\_eSWT

```
labelData = new FormAttachment (100, 0);

setLayoutData(labelData);
FormData progressData = new FormData ();
progressData = new FormAttachment (0, 5);
progressData = new FormAttachment (100, -5);
progressData = new FormAttachment (100, -5);
setLayoutData(progressData);
setMaximum(10);
pack();

Rectangle splashRect = splash.getBounds();
Rectangle displayRect = display.getBounds();
int x = (displayRect.width - splashRect.width) / 2;
int y = (displayRect.height - splashRect.height) / 2;
splash.setLocation(x, y);
pack();

displayExec(new Runnable() {
public void run() {
for (int i = 0; i < 10; i++) {
setSelection(i+1);          bar.
update();                  bar.
try {Thread.sleep(1000);} catch (InterruptedException e) {
}
}

close();          splash.
dispose();       image.
setVisible(true); shell.
}
});

// Execute the eSWT event loop.
while(!exiting) {
if(!display.readAndDispatch()) {
display.sleep();
}
}

// Clean up and destroy the MIDlet.
display.dispose();
notifyDestroyed();
}

public void widgetDefaultSelected(SelectionEvent e) {
}

public void widgetSelected(SelectionEvent e) {
// Exit command selected, exit the event loop.
exitEventLoop();
}
}
```

## Postconditions

The splash screen will appear for 10 secs and update the ProgressBar every second. And will make the main application screen visible.

## See also

- [eSWT](#)