

This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. The article is believed to be still valid for the original topic scope.



## Contents

- [1 Introduction](#)
- [2 Timer Basic](#)
  - ◆ [2.1 One Shot Scheduler](#)
  - ◆ [2.2 Periodic Timer](#)
- [3 Usage Example](#)
- [4 Bouncing Ball Example](#)
  - ◆ [4.1 widget.xml](#)
  - ◆ [4.2 bouncing\\_ball.he](#)
- [5 Code Snippet](#)
- [6 See Also](#)

## Introduction

In WidSets widget, you can set up schedule timer both one shot scheduler and periodic timer. This page will show you how to use them.

## Timer Basic

**schedule** function is used for set up timer scheduler. When executed the system calls **void timerEvent(Timer timer)** function on widget script.

## One Shot Scheduler

```
Timer schedule(long delay)
```

This type of timer, **timerEvent** will be called only once after delay **milliseconds** passed.

## Periodic Timer

```
Timer schedule(long delay, long period)
```

Schedules a timer for repeated fixed-delay execution, beginning after the specified delay. Subsequent executions take place at approximately regular intervals separated by the specified period.

## Usage Example

Define Timer object as class member variable.

```
Timer g_timer;
```

Add Timer object construction in **openWidget** function.

```
Shell openWidget ()
{
    ...
    // Delete timer if existed
    if (g_timer != null)
    {
        g_timer.cancel();
        g_timer = null;
    }
    // Construct timer
    g_timer = schedule(1000, 10);
    ...
}
```

Add **timerEvent** function.

```
void timerEvent(Timer timer)
{
    ...
}
```

This function will be first called after schedule function has been called for 1 second and will be called every 10 milliseconds after that as scheduled in openWidget.

And don't forget to add timer destruction in **stopWidget** function and **before exit widget** or your widget may be has big problem.

```
void stopWidget ()
{
    // Delete timer if existed
    if (g_timer != null)
```

```

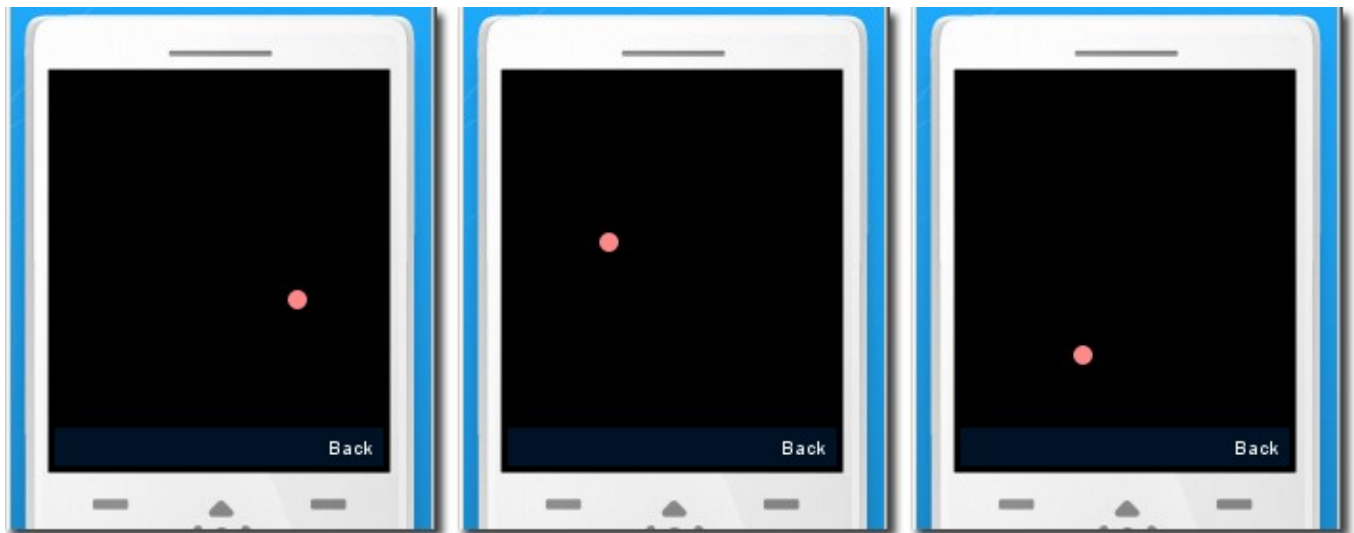
    {
        g_timer.cancel();
        g_timer = null;
    }
}

void actionPerformed(Shell shell, Component source, int action)
{
    //when CMD_BACK event comes in, pop the current shell (this widget)
    if (action == CMD_BACK) {
        // Delete timer if existed
        if (g_timer != null)
        {
            g_timer.cancel();
            g_timer = null;
        }
        popShell(shell);
    }
}
}

```

## Bouncing Ball Example

This example will show you how to use timer.



### widget.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<widget spec_version="2.1">
    <info>
        <name>Bouncing Ball</name>
        <version>1.0</version>
        <author>Sittiphol Phanvilai</author>
        <clientversion>1.0</clientversion>
        <shortdescription>Bouncing Ball (Timer Example)</shortdescription>
        <longdescription>Example show you how to use Timer</longdescription>
        <tags>example timer</tags>
    </info>
</widget>

```

## WidSets\_for\_Rookie\_EP\_9\_: Timer

```
</info>

<parameters>
  <parameter type="string" name="widgetname" description="Name of widget" editable="no">Bound
</parameters>

<resources>
  <code src="bouncing_ball.he"/>

  <stylesheet>
    bg {
      color-1: white;
      background: solid green;
      align: hcenter vcenter;
      border: 1 1 1 1;
      border-type: rectangle white;
    }

    cenmid
    {
      align: hcenter vcenter;
    }

    canvas {
      color-1: black;
      color-2: red;
      color-3: white;
      font-3: small bold underlined;
    }
  </stylesheet>

  
</resources>

<layout minimizedheight="45sp">
  <view id="viewMini">
    
  </view>

  <view id="viewMaxi" class="bg">
    <script id="hello" class="text"/>
  </view>

  <webview>
    <weblabel class="top: 0px; left: 10px;" style="color: black;">${widgetname}</weblabel>
  </webview>
</layout>

</widget>
```

### bouncing\_ball.he

```
/*
 * Copyright (C) 2008 Nokia Corporation.
 *
 * Licensed under separate Nokia Corporation End-user Software
 * Agreement (the "License").
 *
 * You may not use this Software except in compliance with the License.
 * The Software is distributed under the License on "AS IS" basis,
```

## WidSets\_for\_Rookie\_EP\_9\_:Timer

```
* withouth warranties. See the License for the specific language
* governing rights and limitations under the License.
*
* Modified by Sittiphol Phanvilai (Neois)
*
*/
```

```
class bouncing_ball
{
    //It's nice to store command ids to static constants
    const int CMD_BACK = 1;

    //MenuItems are displayed over phone's soft buttons
    //Usually to go back, ok, open options menu etc
    MenuItem BACK = new MenuItem(CMD_BACK, "Back");

    Shell g_shell = null;
    Canvas g_canvas = null;
    Timer g_timer;

    int g_BallX = 30;
    int g_BallY = 20;

    int g_incX = 1;
    int g_incY = 1;

    //WidSets framework will call createElement() per
    //each script-element it finds from views being
    //created by createView()

    Component createElement(String viewName,
                            String elementName,
                            Style style,
                            Object context)
    {
        return null;
    }

    void startWidget()
    {
        //instantiate minimized view in startup
        setMinimizedView(createView("viewMini", getStyle("bg")));
    }

    Shell openWidget()
    {
        //instantiate maximized view when user opens this widget
        g_canvas = new Canvas(getStyle("canvas"));
        final Shell shell = new Shell(g_canvas);

        //place canvas over automatic Scrollable created by Shell
        shell[0] = shell[0][0];

        cancelTimer();
        g_timer = schedule(10, 10);

        return (g_shell = shell);
    }

    void stopWidget()
    {
        cancelTimer();
    }
}
```

## WidSets\_for\_Rookie\_EP\_9:\_Timer

```
}

MenuItem getSoftKey(Shell shell, Component focused, int key)
{
    //return the key we want to display at position=SOFTKEY_BACK
    //this usually is the Right Soft Button (RSB), for other key
    //positions return null, as we don't want other keys
    if (key == SOFTKEY_BACK) {
        return BACK;
    }
    return null;
}

void actionPerformed(Shell shell, Component source, int action)
{
    //when CMD_BACK event comes in, pop the current shell (this widget)
    if (action == CMD_BACK) {
        cancelTimer();
        popShell(shell);
    }
}

void paint(Component c, Graphics g, Style style, int width, int height)
{
    g.setColor(style.color(0));
    g.fillRect(0, 0, width, height);
    g.setColor(0xFF8888);
    g.fillArc(g_BallX, g_BallY, 10, 10, 0, 360);
}

void timerEvent(Timer timer)
{
    int w, int h = g_canvas.getSize();
    if (g_BallX == 0 || g_BallX == w-10)
        g_incX = -g_incX;
    if (g_BallY == 0 || g_BallY == h-10)
        g_incY = -g_incY;
    g_BallX += g_incX;
    g_BallY += g_incY;
    g_shell.repaint(false);
    flushScreen(true);
}

void cancelTimer()
{
    if (g_timer != null)
    {
        g_timer.cancel();
        g_timer = null;
    }
}
}
```

## Code Snippet

You can download source code for this tutorial from [File:WidSets Canvas Example.zip](#)

## See Also

- [WidSets for Rookie EP 1 : First Step to WidSets SDK](#)
- [WidSets for Rookie EP 2 : First Compilation with WidSets SDK](#)
- [WidSets for Rookie EP 3 : Understand Hello World](#)
- [WidSets for Rookie EP 4 : Fasten WidSets Development](#)
- [WidSets for Rookie EP 5 : EditPlus Integration](#)
- [WidSets for Rookie EP 6 : Softkey Menu](#)
- [WidSets for Rookie EP 7 : Standard UI](#)
- [WidSets for Rookie EP 8 : Canvas](#)
- **WidSets for Rookie EP 9 : Timer**
- [WidSets for Rookie EP 10 : Key Handling](#)
- [WidSets for Intermediate EP 1 : HTTP Request](#)
- [WidSets for Intermediate EP 2 : HTTP with XML Filter](#)
- [WidSets for Advance EP 1 : Life Pictures Project](#)
- [WidSets SDK Tips : Emulator Language Changing](#)
- [WidSets SDK Tips : Emulator Skin Changing](#)
- [WidSets SDK Tips : Add Custom Emulator Skin](#)