

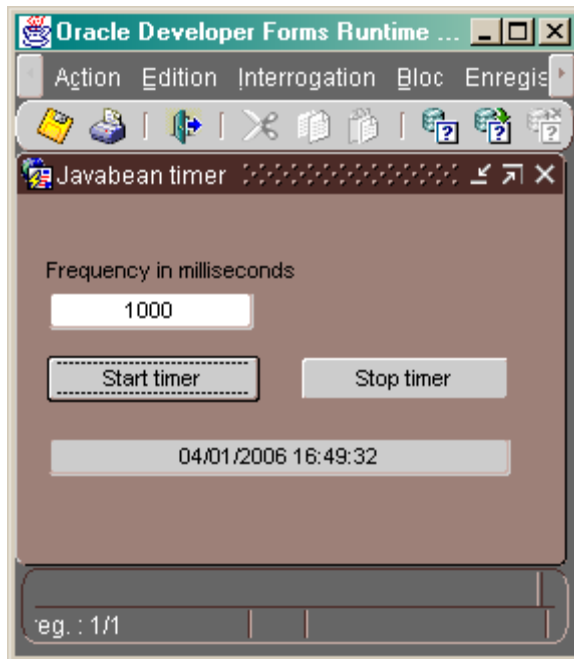
Oracle Forms – Javabeen

A Javabeen timer

[Home page](#)

1. Purpose

This is a Javabeen component that allows to replace the internal Oracle Forms' timer



2. The Javabeen

```

package oracle.forms.demo;

import oracle.forms.handler.IHandler;
import oracle.forms.ui.CustomButton;
import oracle.forms.properties.ID;
import oracle.forms.ui.VBean;

import oracle.forms.engine.Main;
import oracle.forms.engine.*;
import oracle.forms.handler.*;

public class Timer extends VBean implements Runnable
{
    static Thread runner ;
    static int seconds = 0 ;
    static IHandler mHandler;
    protected static final ID TIMEREXPIRED = ID.registerProperty("TimerExpired");
    protected static final ID pInitTimer = ID.registerProperty("initTimer");
    protected static final ID pStartTimer = ID.registerProperty("Start");
    protected static final ID pStopTimer = ID.registerProperty("Stop");

    public Timer()
    {
        super();
    }
}

```

```

}

public void init(IHandler handler)
{
    super.init(handler);
    mHandler = handler;
    System.out.println("*** oracle.forms.demo.Timer Init ***") ;
}

private void startTimer()
{
    if (runner == null )
    {
        runner = new Thread(this);
        runner.start();
    }
}

private static void stopTimer()
{
    if (runner != null )
    {
        runner = null;
    }
}

public void run()
{
    Thread theThread = Thread.currentThread();
    while (runner == theThread)
    {
        try{
            Thread.sleep(seconds);
        } catch (InterruptedException e) { }
        dispatch_event() ;
    }
}

public boolean setProperty(ID _ID, Object _args)
{
    if(_ID==pInitTimer)
    {
        System.out.println("milliseconds=" + (String)_args) ;
        seconds = new Integer((String)_args).intValue() ;
        return true;
    }

    else if (_ID == pStartTimer)
    {
        System.out.println("Start") ;
        startTimer() ;
        return true;
    }

    else if (_ID == pStopTimer)
    {
        stopTimer() ;
        return true;
    }
    else
    {
        return true;
    }
}

public void dispatch_event()
{
    CustomEvent ce = new CustomEvent(mHandler, TIMEREXPIRED);
    dispatchCustomEvent(ce);
}
}

```

3. Forms configuration

- Copy the **bean_timer.jar** file in the /forms/java directory

- Edit the **/forms/server/formsweb.cfg** file to add the jar file to the **archive_jini** variable

```
archive_jini=f90all_jinit.jar,.....,bean_timer.jar
```

4. How to implement this bean in your own form

- Open your form
- Add a Javabean component to any block
- Set its **Implementation class property** to : **oracle.forms.demo.Timer**

5. The properties that can be sent to the bean

- **The frequency of expiration**

```
set_custom_property(
'BLOC3.BEAN_ITEM', 1,
'initTimer',
'number_of_milliseconds');
```

- **Start the timer**

```
set_custom_property(
'BLOCK.BEAN', 1, 'Start',
'');
```

- **Stop the timer**

```
set_custom_property(
'BLOCK.BEAN', 1, 'Stop', '');
```

6. The sample dialog

- [Download the beantimer.zip file](#)
- Unzip the **beantimer.zip** file
- Copy the **bean_timer.jar** file in your **/forms/java/** directory
- Edit your **/forms/server/formsweb.cfg** file
- Open the **bean_timer.fmb** module (Oracle Forms 9.0.2)
- Compile all and run the module

This dialog allows to enter the frequency of the timer, start (showing the time in real-time) and stop the timer.

The code that captures the java event is located in the **WHEN-CUSTOM-ITEM-EVENT** trigger of the bean item.

```
:BLOCK2.HEURE := SYSDATE ;  
Synchronize ;
```