

# SimpleTimer, ver. 2.0.3

[\[Properties\]](#) [\[Methods\]](#) [\[Events\]](#) [\[Stand-alone Methods\]](#)  
[\[How To Use\]](#) [\[Known Bugs\]](#) [\[Comments\]](#)

SimpleTimer is a timer class. It has the same timer resolution as TTimer, but it is more lightweight because it's derived from TObject in stead of TComponent. Furthermore, the same handle is shared between multiple instances of SimpleTimer. This makes it ideal for developers who need a timer in their own components or applications, but want to keep the resource usage minimal.

## Properties

Enabled	<b>property Enabled: Boolean;</b> Enables (starts) or disables (stops) the timer.	Default False
Interval	<b>property Interval: Cardinal;</b> The interval of the timer in millisecs. <b>NOTE:</b> Specifying a value of 0 will cause the timer to stop, but will not set the Enabled property to false.	Default 1000
AutoDisable	<b>property AutoDisable: Boolean;</b> If true, the timer will disable itself (Enabled is set to False) immediately after the next time it fires. Useful when you want a one-shot timer.	Default False

## Methods

Create	<b>constructor Create;</b> Creates a new TSimpleTimer object.
CreateEx	<b>constructor CreateEx(AInterval: Cardinal; AOnTimer: TNotifyEvent);</b> Creates a new TSimpleTimer object with the specified Interval property and OnTimer event.
Destroy	<b>destructor Destroy; override;</b> Destroys the TSimpleTimer object.

## Events

OnTimer	<b>property OnTimer: TNotifyEvent;</b> Called when the timer fires.
---------	--

## Stand-alone Methods

GetSimpleTimerCount	<b>function GetSimpleTimerCount: Cardinal;</b> Returns the number of TSimpleTimer objects currently allocated.
GetSimpleTimerActiveCount	<b>function GetSimpleTimerActiveCount: Cardinal;</b> Returns the number of TSimpleTimer objects that are currently active (enabled).

## How To Use

This example creates a SimpleTimer object, initializing Interval to 500 milliseconds and the OnTimer event to TForm1.TimerProc1, then starts the timer:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    SimpleTimer1 := TSimpleTimer.CreateEx(500, TimerProc1);
```

```
SimpleTimer1.Enabled := True;  
end;
```

This is what TimerProc1 might look like:

```
procedure TForm1.TimerProc1(Sender: TObject);  
begin  
  ListBox1.Items.Add('SimpleTimer1 fired');  
end;
```

Remember to destroy the SimpleTimer object when your app. terminates.

## Known Bugs

- You may experience occasionally that you lose a timer event when you work with small intervals (approx. below 100 ms.). This is normal, and the same thing happens with a TTimer object. If you want precision you need a multimedia timer, not TTimer or TSimpleTimer which use the low-priority WM\_TIMER message. WinXP has better handling of timers and will generally lose fewer timer events than other Windows versions.

## Comments

SimpleTimer is *free for personal and commercial use*. Feel free to use and improve it, but *please include all original files if you redistribute the zip-file*. If you have any comments or corrections I would very much like to hear them.

Get the latest version from <http://subsimple.com/delphi.asp>.

Troels Jakobsen

[troels.jakobsen@gmail.com](mailto:troels.jakobsen@gmail.com)