

Using the FastTimer Component

See also [TFastTimer Reference](#) [Example](#)

Purpose

Use the FastTimer component to trigger an event, either one time or repeatedly, after a measured interval. You write the code that you want to occur at the specified time inside the timer component's [OnTimer](#) event. Unlike the built-in Timer component, which is limited to 18.2 ticks/second, the FastTimer component has a resolution selectable down to 1ms.

Tasks

To specify the amount of elapsed time before the timer event is triggered, use the [Interval](#) property.

To start a timed event, set the fasttimer components Enabled property to True. To discontinue a timed event, set the fasttimer component's Enabled property to False.

To change the granularity of the timer, use the [Resolution](#) property. Be aware that resolutions of less than 10ms may eat up lots of CPU time.

Additional Notes

When the FastTimer is set to generate OnTimer events at very high rates, it is best to limit the amount of code executed during the OnTimer event.

Unlike the built-in Timer component, a FastTimers Enabled property defaults to FALSE.

See also

Common Component Tasks

Example

This following example implements a simple one-minute cycling digital clock.

```
procedure TForm1.FormActivate(Sender: TObject);  
begin  
    with FastTimer1 do  
        begin  
            Interval := 1000; { set for one second delay }  
            Enabled := True;   { enable, since default is disabled }  
        end;  
end;  
  
const  
    Seconds : Integer = 0;  
  
procedure TForm1.FastTimer1Timer(Sender: TObject);  
begin  
    Inc(Seconds);  
    if Seconds := 60  
        then  
            Seconds := 0;  
    with Label1 do  
        begin  
            Caption := IntToStr(Seconds);  
            Update;  
        end;  
end;
```

TFastTimer Component

[Properties](#) [Events](#) [Tasks](#)

Unit

FastTime

Description

The TFastTimer component causes an [OnTimer](#) event to occur whenever a specified period of time passes. Within that OnTimer event handler, your code specifies what you want to happen each time the OnTimer event occurs. The TFastTimer has a higher potential resolution than the built-in TTimer, which is limited to 55ms (18.2 times/second).

You use the [Interval](#) property to control the amount of time between timer events.

To activate or deactivate a timer, use its Enabled property.

To change the granularity of timer, set its [Resolution](#) property.

In addition to these properties and events, this component also has the properties and methods that apply to all components.

Properties

ComponentIndex

Interval

Owner

Tag

Enabled

Name

Resolution

Events

[OnTimer](#)

OnTimer Event

See also [Example](#)

Applies to

[TFastTimer](#) component

Declaration

property OnTimer: TNotifyEvent;

Description

The OnTimer event is used to execute code at regular intervals. Place the code you want to execute within the OnTimer event handler.

The [Interval](#) property of a fasttimer component determines how frequently the OnTimer event occurs. Each time the specified interval passes, the OnTimer event occurs.

Additional Notes

When the FastTimer is set to generate OnTimer events at very high rates, it is best to limit the amount of code executed during the OnTimer event.

See also

[Interval](#) property

Example

This example demonstrates how a FastTimer might be used to control updates of frames in a game. In this example, Run1 and Stop1 are items on a menu. DisplayOK is a boolean flag that tells the main program when it is OK to send an updated frame to the screen. StopIt is a boolean flag that tells the main program to end.

```
procedure TForm1.Run1Click(Sender: TObject);  
begin  
    StopIt := False;  
    DisplayOK := False;  
    FastTimer1.Interval := 30; { Tick 33.3 times/sec }  
    FastTimer1.Enabled := True;  
  
    while not StopIt do  
        begin  
  
            { Do processing to create offscreen buffer }  
  
            repeat  
                Application.ProcessMessages { Cooperate with Windows }  
            until DisplayOK;  
            DisplayOK := False;  
  
            { Move offscreen buffer onscreen }  
  
        end;  
        FastTimer1.Enabled := False;  
end;  
  
procedure TForm1.FastTimer1Timer(Sender: TObject);  
begin  
    DisplayOK := True;  
end;  
  
procedure TForm1.Stop1Click(Sender: TObject);  
begin  
    StopIt := True;  
end;  
  
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);  
begin  
    StopIt := True;  
end;
```

Interval Property

[See also](#)

Applies to

[TFastTimer](#) component

Declaration

property Interval: Word;

Description

The Interval property determines in milliseconds the amount of time that passes before the fasttimer component initiates another [OnTimer](#) event.

You can specify any value between 0 and 65,535 as the interval value, but the fasttimer component won't call an OnTimer event if the value is 0. The default value is 1000 (one second).

The actual timing of OnTimer events may be affected by the value of the [Resolution](#) property.

See also

[Resolution](#) property

Resolution Property

[See Also](#)

Applies to

[TFastTimer](#) component

Declaration

property Resolution: Word;

Description

The Resolution property determines in milliseconds the precision of the timers [Interval](#). Setting a small resolution will make the timer more accurate, but it will also eat up more CPU time. The built-in TTimer has a fixed resolution of 55 ms (1/18.2 second).

You can specify any value between 1 and 65,535 as the resolution value, but the fasttimer component won't call an [OnTimer](#) event if the value is 0. The default value is 10 (10 ms, or 1/100 sec). On some machines, some resolutions might not be available (particularly smaller values). The fasttimer component will select the closest available resolution for the machine if this is the case. If precision is necessary, check the Resolution value immediately after setting it.

See also

[Interval](#) property

