

***fast*Timer**

A high-resolution timer component for Delphi™

Copyright © 1995 Eric Nielsen All Rights Reserved

Table of Contents

Introduction.....	3
Shareware	
Disclaimer - Agreement.....	4
Installing the FastTimer component.....	5
Using the FastTimer component.....	6
TFastTimer component.....	7
Interval Property.....	8
Resolution Property.....	9
OnTimer Event.....	10
Sample Code.....	11
Ordering Information.....	13
Order Form.....	15

Introduction

Borland's Delphi™ is a great tool for rapid development of Windows applications. It provides several ready-made components that make much of the development process a mere drag-and-drop activity.

The included Timer component, however, is not suitable for some applications, since its effective resolution is only 55ms (18.2 ticks/second). The FastTimer component is designed to be a drop-in solution for this problem. Although its use is nearly identical to that of the built-in Timer, FastTimer provides up to 1ms (1000 ticks/second) resolution, and it should prove to be useful for applications such as screen savers, games, or any other type of application that requires regular timing events.

Shareware

Shareware distribution gives users a chance to try software before buying it. If you try a Shareware program and continue using it, you are expected to register. Individual programs differ on details -- some request registration while others require it, some specify a maximum trial period. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

Copyright laws apply to both Shareware and commercial software, and the copyright holder retains all rights, with a few specific exceptions as stated below. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. (In both cases, there are good programs and bad ones!) The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all and sundry or to a specific group. For example, some authors require written permission before a commercial disk vendor may copy their Shareware.

Shareware is a distribution method, not a type of software. You should find software that suits your needs and pocketbook, whether it's commercial or Shareware. The Shareware system makes fitting your needs easier, because you can try before you buy. And because the overhead is low, prices are low also. Shareware has the ultimate money-back guarantee -- if you don't use the product, you don't pay for it.

Disclaimer - Agreement

Users of FastTimer Component for Delphi must accept this disclaimer of warranty: "FastTimer Component for Delphi is supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of FastTimer Component for Delphi."

FastTimer Component for Delphi is a "shareware program" and is provided at no charge to the user for evaluation. Feel free to share it with your friends, but please do not give it away altered or as part of another system. The essence of "user-supported" software is to provide personal computer users with quality software without high prices, and yet to provide incentive for programmers to continue to develop new products. If you find this program useful and find that you are using FastTimer Component for Delphi and continue to use FastTimer Component for Delphi after a reasonable trial period, you must make a registration payment to Eric Nielsen. Registration fees are explained on the order form. The registration fee will license one copy for use on any one computer at any one time. You must treat this software just like a book. An example is that this software may be used by any number of people and may be freely moved from one computer location to another, so long as there is no possibility of it being used at one location while it's being used at another. Just as a book cannot be read by two different persons at the same time. Certain portions of the FastTimer Component for Delphi may be redistributed royalty-free by registered users of the FastTimer Component for Delphi with products that use the Component. These portions include FASTTIME.DLL and any code linked from FASTTIME.DCU, and, in the case of source-code registered users, any code compiled and linked from the source-code whether original or modified.

Commercial users of FastTimer Component for Delphi must register and pay for their copies of FastTimer Component for Delphi within 30 days of first use or their license is withdrawn. See the order form for Site-License arrangements.

The evaluation version of FastTimer Component for Delphi may be distributed freely by shareware and CD-ROM vendors, and through on-line services such as BBS's and other systems, provided that all the files listed below (and only the files listed below) are included as part of the FastTimer Component for Delphi package, and that they remain intact and unchanged. Please consider contacting Eric Nielsen at the address below so that you will the latest version.

You are encouraged to pass a copy of FastTimer Component for Delphi along to your

friends and colleagues for evaluation. Please encourage them to register their copy if they find that they can use it. All registered users will receive a copy of the latest version of the FastTimer Component for Delphi.

Installing the FastTimer component

Install the files

First copy the following files into the directory where you keep your additional (non-Borland) components:

FASTTIME.DCU
FASTTIME.DCR

Then copy the following file to your WINDOWS directory:

FASTTIME.DLL

And put these two in your DELPHI\BIN directory:

FASTTIME.HLP
FASTTIME.KWF

Install the help reference

In your Delphi program group, start the program called HelpInst. Open the file DELPHI.HDX (in your DELPHI\BIN directory). From the Keyword menu, select "Add Keyword File" and select FASTTIME.KWF. From the "File" menu, select "Save". Then exit HelpInst.

Install the component

Start Delphi and select "Install Components" from the "Options" menu. Click the add button, then browse to find your component directory. Click on FASTTIME.DCU, then OK, and your library will be rebuilt. The FastTimer component will be placed on the palette in the "New" section. The installation is now complete.

Using the FastTimer Component

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 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 FastTimer's Enabled property defaults to FALSE.

TFastTimer Component

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

Enabled

Interval

Name

Owner

Resolution

Tag

Events

OnTimer

Interval Property

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

Applies to

TFastTimer component

Declaration

property Resolution: Word;

Description

The Resolution property determines in milliseconds the precision of the timer's 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

OnTimer Event

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

Sample Code

Example 1

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;
```

Example 2

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;
```

Ordering Information

To register FastTimer, send \$10.00 + \$4.95 for shipping (\$6.95 for shipping outside the United States) along with your address. If you would like the source code as well, send \$15.00 + \$4.95 for shipping (\$6.95 outside the United States). Site licenses are \$75.00. All site licenses have the source code included.

For your convenience, an order form is included below.

Want to save shipping costs?

We can send the package via e-mail on CompuServe, America Online, and the Internet. It saves us, so it saves you! Please include your full mailing address with the order.

Order by CompuServe: GO SWREG

Register #6873 for FastTimer, #6874 for FastTimer + Source, or #6875 for a FastTimer Site License.

Credit card orders ONLY

You can order with MC, Visa, Amex, or Discover from Public (software) Library by calling 800-2424-PsL or 713-524-6394 or by FAX to 713-524-6398 or by CIS Email to 71355,470. You can also mail credit card orders to PsL at P.O.Box 35705, Houston, TX 77235-5705.

When you contact PsL, ask for Item/Product #14162.

THE ABOVE NUMBERS ARE FOR CREDIT CARD ORDERS ONLY. THE AUTHOR OF THIS PROGRAM CANNOT BE REACHED AT THESE NUMBERS.

Any questions about the status of the shipment of the order, refunds, registration options, product details, technical support, volume discounts, dealer pricing, site licenses, non-credit card orders, etc., must be directed to:

Eric Nielsen
RR 1 Box 420
Alto Pass IL, 62905
USA

CompuServe: 75270,3070
America Online: htrsoft

Internet: htrsoft@midwest.net

To insure that you get the latest version, PsL will notify us the day of your order and we will ship the product directly to you.

International orders

We prefer that you order by credit card, but we will also accept hard currency. If you have any special shipping requirements, be sure to let us know!

Licensing and Distribution

You may distribute FASTTIME.DLL with your applications royalty-free. You may not distribute the registered FASTTIME.DLL without your application (except as an update for an already installed program). If several developers will be using FASTTIME, each should register a copy of FASTTIME for their machine, or you should purchase a site license.

Support and Updates

If you find a bug in this product, please let us know. We can be reached by phone, by mail, and by E-mail. Registered users can request to be notified of updates by email.

Contact Information

Mail: Eric Nielsen
RR 1 Box 420
Alto Pass IL, 62905
USA

Phone: (618) 893-2244

CompuServe: 75270,3070
America Online: htrsoft
Internet: htrsoft@midwest.net

Order Form

This order form is for checks/money orders ONLY. If you would like to pay by credit card, please see Ordering Information above.

Company: _____

Name: _____

Address: _____

City: _____ State/

Province: _____

Zip: _____ Country: _____

CompuServe Id: _____

Email: _____

Where did you find FastTimer? _____

<u>Product(s)</u>	<u>Quantity</u>	<u>Price</u>	<u>Total</u>
FastTimer component	_____x	\$ 10.00	\$ _____
FastTimer component + source	_____x	\$ 15.00	\$ _____
FastTimer site license	_____x	\$ 75.00	\$ _____
Shipping (\$4.95 U.S., \$6.95 outside the U.S.)			\$ _____
		TOTAL:	\$ _____