

IGLTime v1.0

Crossplatform Timing class

**intended for measuring time
on Linux and Windows
without code change**

CROSSPLATFORM EDITION

For Kylix2 & Delphi6

The IGLTime is free software.

Copyright (C) 2002 Artur Bać All Rights Reserved.

REDA POLAND arturbac@poczta.onet.pl

1. Legal notice

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL ARTUR BAĆ BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Developed with Kylix2 Open Edition & Delphi6 Open Edition

**Piece of part of windows code is written on the base
of source CGLib project by Tom Nuygen**

2. Requirements

Tested on Mandrake 8.1 and Windows XP

2.1. Linux users

a) Linux OS (ex. Mandrake 8.0 , Redhat 7.1)

b) Kylix2 instaled

For Mandrake 8.1 uses

I'm awaitng for patch for borlands qtinf library thus there is a problem

Kylix uses Qt library ver 2.3.0 , but Mandrake 8.1 Qt lib ver 2.3.1

As I read in Qt 2.3.1 notes they have to be compatible

But this qtinf lib make exceptions and will not work with qt 2.3.1

Problem we can obey with simple script that will start our compiled Application

Just remake copy of starkylix script to point your compiled App.

2.2. Windows users

Windows 98, Me, NT, 2000, XP operating system
Delphi 6 installed

3. Using

Using is identical for both Linux and Windows systems.

3.1. Types

```
__time_t      = longint;
__suseconds_t = longint;
```

timeval = record

```
tv_sec : __time_t;      { Seconds. }
tv_usec : __suseconds_t; { Microseconds. 10^-6 sec }
end;
```

3.2. Methods

Constructor Create;

.....hymm for creating variable

Procedure Start;

This method will start timing (will remember time of start)

Procedure Stop;

This method will stop timing (will remember time of end)

Procedure Reset;

This method will reset remembered values to 0;
Is called automaticaly on start.

3.3. Properties

Property ErrorMessage : ansistring Read mess;

This property tells us if something is wrong
In normal state will say 'Timer ok'
on error appropriate error text

Property TimeElapsed : timeval Read GetTimeElapsed;

Returns time elapsed since start in timeval type
if timer is not stopped will be stopped automaticaly.

Property SecElapsed : int64 Read GetSecElapsed;

Returns time elapsed in seconds since start in int64 type
if timer is not stopped will be stopped automaticaly.

Property USecElapsed : int64 Read GetUSecElapsed;

Returns time elapsed in microseconds since start in int64 type
if timer is not stopped will be stopped automaticaly.

3.4. Initialization

example

```
.....  
        { Private declarations of form class}  
        PrecTimer : IGLTimer;  
.....
```

Creation of form class

```
.....  
PrecTimer := IGLTimer.Create;  
.....  
//Redy to use
```

anywhere you need

```
.....  
PrecTimer2.Start;  
.....  
PrecTimer2.Stop;  
        label.Caption:=Format('Code execution took %10d usec ',[PrecTimer2.USecElapsed]);  
.....
```

End