

PASCAL - DATE AND TIME

http://www.tutorialspoint.com/pascal/pascal_date_time.htm

Copyright © tutorialspoint.com

Most of the softwares you write need implementing some form of date functions returning current date and time. Dates are so much part of everyday life that it becomes easy to work with them without thinking. Pascal also provides powerful tools for date arithmetic that makes manipulating dates easy. However, the actual name and workings of these functions are different for different compilers.

Getting the Current Date & Time

Pascal's TimeToString function gives you the current time in a colon: delimited form. The following example shows how to get the current time –

```
program TimeDemo;
uses sysutils;

begin
  writeln ('Current time : ',TimeToStr(Time));
end.
```

When the above code was compiled and executed, it produces the following result –

```
Current time : 18:33:08
```

The **Date** function returns the current date in **TDateTime** format. The TDateTime is a double value, which needs some decoding and formatting. The following program demonstrates how to use it in your program to display the current date –

```
Program DateDemo;
uses sysutils;
var
  YY,MM,DD : Word;

begin
  writeln ('Date : ',Date);
  DeCodeDate (Date,YY,MM,DD);
  writeln (format ('Today is (DD/MM/YY): %d/%d/%d ',[dd,mm,yy]));
end.
```

When the above code was compiled and executed, it produces the following result –

```
Date: 4.1113000000000000E+004
Today is (DD/MM/YY):23/7/2012
```

The Now function returns the current date and time –

```
Program DatenTimeDemo;
uses sysutils;
begin
  writeln ('Date and Time at the time of writing : ',DateTimeToStr(Now));
end.
```

When the above code was compiled and executed, it produces the following result –

```
Date and Time at the time of writing : 23/7/2012 18:51:
```

Free Pascal provides a simple time stamp structure named **TTimeStamp**, which has the following format –

```
type TTimeStamp = record
```

```

Time: Integer;
Date: Integer;
end;

```

Various Date & Time Functions

Free Pascal provides the following date and time functions –

Sr.No.	Function Name & Description
1	function DateTimeToFileDate <i>DateTime: TDateTime; LongInt;</i> Converts DateTime type to file date.
2	function DateTimeToStr <i>DateTime: TDateTime;;</i> Constructs string representation of DateTime
3	function DateTimeToStr <i>DateTime: TDateTime; constFormatSettings: TFormatSettings;;</i> Constructs string representation of DateTime
4	procedure DateTimeToString <i>outResult: ; constFormatStr: ; constDateTime: TDateTime;</i> Constructs string representation of DateTime
5	procedure DateTimeToString <i>outResult: ; constFormatStr: ; constDateTime: TDateTime; constFormatSettings: TFormatSettings;</i> Constructs string representation of DateTime
6	procedure DateTimeToSystemTime <i>DateTime: TDateTime; outSystemTime: TSystemTime;</i> Converts DateTime to system time
7	function DateTimeToTimeStamp <i>DateTime: TDateTime; TTimeStamp;</i> Converts DateTime to timestamp
8	function DateToStr <i>Date: TDateTime;;</i> Constructs string representation of date
9	function DateToStr <i>Date: TDateTime; constFormatSettings: TFormatSettings;;</i> Constructs string representation of date
10	function Date: TDateTime; Gets current date

11 **function DayOfWeekDateTime: TDateTime:Integer;**
Gets day of week

12 **procedure DecodeDateDate: TDateTime; outYear: Word; outMonth: Word; outDay: Word;**
Decodes DateTime to year month and day

13 **procedure DecodeTime**
Time: TDateTime; outHour: Word; outMinute: Word; outSecond: Word; outMilliSecond: Word;
Decodes DateTime to hours, minutes and seconds

14 **function EncodeDateYear: Word; Month: Word; Day: Word:TDateTime;**
Encodes year, day and month to DateTime

15 **function EncodeTimeHour: Word; Minute: Word; Second: Word; MilliSecond: Word**
:TDateTime;
Encodes hours, minutes and seconds to DateTime

16 **function FormatDateTimeconstFormatStr: ; DateTime: TDateTime;;**
Returns string representation of DateTime

17 **function FormatDateTime**
constFormatStr: ; DateTime: TDateTime; constFormatSettings: TFormatSettings;;
Returns string representation of DateTime

18 **function IncMonthconstDateTime: TDateTime; NumberOfMonths: Integer = 1:TDateTime;**
Adds 1 to month

19 **function IsLeapYearYear: Word:Boolean;**
Determines if year is leap year

20 **function MSecsToTimeStampMSecs: Comp:TTimeStamp;**
Converts number of milliseconds to timestamp

21 **function Now: TDateTime;**
Gets current date and time

22

function StrToDateTime*const S: TDateTime;*

Converts string to DateTime

23

function StrToDateTime*const S: ShortString; const FormatSettings: TFormatSettings*
:TDateTime;

Converts string to DateTime

24

function StrToDateTime*const S: AnsiString; const FormatSettings: TFormatSettings*
:TDateTime;

Converts string to DateTime

25

function StrToDate*const S: ShortString;***TDateTime;**

Converts string to date

26

function StrToDate*const S: Ansistring;***TDateTime;**

Converts string to date

27

function StrToDate*const S: ShortString; separator: Char;***TDateTime;**

Converts string to date

28

function StrToDate*const S: AnsiString; separator: Char;***TDateTime;**

Converts string to date

29

function StrToDate*const S: ShortString; const useformat: ; separator: Char;***TDateTime;**

Converts string to date

30

function StrToDate*const S: AnsiString; const useformat: ; separator: Char;***TDateTime;**

Converts string to date

31

function StrToDate*const S: PChar; Len: Integer; const useformat: ; separator: Char = #0*
:TDateTime;

Converts string to date

32

function StrToTime*const S: Shortstring;***TDateTime;**

Converts string to time

33

function StrToTime*const S: Ansistring;***TDateTime;**

Converts string to time

34

function StrToTime*const S: ShortString; separator: Char: TDateTime;*

Converts string to time

35

function StrToTime*const S: AnsiString; separator: Char: TDateTime;*

Converts string to time

36

function StrToTime*const S: ; FormatSettings: TFormatSettings: TDateTime;*

Converts string to time

37

function StrToTime*const S: PChar; Len: Integer; separator: Char = #0: TDateTime;*

Converts string to time

38

function SystemTimeToDateTime*const SystemTime: TSystemTime: TDateTime;*

Converts system time to datetime

39

function TimeStampToDateTime*const TimeStamp: TTimeStamp: TDateTime;*

Converts time stamp to DateTime

40

function TimeStampToMSecs*const TimeStamp: TTimeStamp: comp;*

Converts Timestamp to number of milliseconds

41

function TimeToStrTime: TDateTime;;

Returns string representation of Time

42

function TimeToStrTime: TDateTime; const FormatSettings: TFormatSettings;;

Returns string representation of Time

43

function Time: TDateTime;

Get current time

The following example illustrates the use of some of the above functions –

```
Program DatetimeDemo;  
uses sysutils;  
var  
year, month, day, hr, min, sec, ms: Word;
```

```

begin
    writeln ('Date and Time at the time of writing : ',DateTimeToStr(Now));
    writeln('Today is ',LongDayNames[DayOfWeek(Date)]);
    writeln;
    writeln('Details of Date: ');

    DecodeDate(Date,year,month,day);
    writeln (Format ('Day: %d',[day]));
    writeln (Format ('Month: %d',[month]));
    writeln (Format ('Year: %d',[year]));
    writeln;
    writeln('Details of Time: ');

    DecodeTime(Time,hr, min, sec, ms);
    writeln (format('Hour: %d',[hr]));
    writeln (format('Minutes: %d',[min]));
    writeln (format('Seconds: %d',[sec]));
    writeln (format('Milliseconds: %d',[hr]));
end.

```

When the above code was compiled and executed, it produced the following result:

```

Date and Time at the time of writing : 7/24/2012 8:26:
Today is Tuesday
Details of Date:
Day:24
Month:7
Year: 2012
Details of Time:
Hour: 8
Minutes: 26
Seconds: 21
Milliseconds: 8
Processing math: 100%

```