TAdrockDates Contents

Help File last updated: 26th March 1996

The methods described in this help file are methods of the class TAdrockDates.

Routines that end with an **i** in the name work with integer variables, those with the same name but without the **i** work with TDateTime variables.

Function ReturnDelphiDayOfWeek(StartingDayOfWeek, WeekDay : Integer) : Integer;
Function ReturnWeekDayName(StartingDayOfWeek, WeekDay : Integer) : String;
Function ReturnWeekofMonth(StartingDayOfWeek : Integer; WorkDate : TDateTime) : Word;
Function ReturnWeekofYear(StartingDayOfWeek : Integer; WorkDate : TDateTime) : Word;
Function GetFirstofMonthWeekNumber(StartingDayOfMonth : Integer; WorkDate : TDateTime) : Integer;

Function ReturnDayOfWeek(StartingDayOfWeek : Integer; WorkDate : TDateTime) : Word;
Function ReturnDayOfMonth(WorkDate : TDateTime) : Word;
Function ReturnDayOfYear(WorkDate : TDateTime) : Word;

Function ReturnMonth(WorkDate : TDateTime) : Word;
Function ReturnQuarter(WorkDate : TDateTime) : Word;
Function ReturnYear(WorkDate : TDateTime) : Word;
Function ReturnCentury(WorkDate : TDateTime) : Word;
Function ReturnDayTh(WorkDate : TDateTime) : String;
Function ReturnDayThi(WorkDay : Integer) : String;

Function ReturnDaysAgo(WorkDate : TDateTime) : String;
Function ReturnWeeksAgo(StartingDayOfWeek : Integer; WorkDate : TDateTime) : String;
Function ReturnMonthsAgo(WorkDate : TDateTime) : String;
Function ReturnQuartersAgo(WorkDate : TDateTime) : String;
Function ReturnYearsAgo(WorkDate : TDateTime) : String;

Function AreDatesInSameMonth(FirstDate, SecondDate : TDateTime) : Boolean;
Function AreDatesInSameYear(FirstDate, SecondDate : TDateTime) : Boolean;
Function AreDatesInSameMonthAndYear(FirstDate, SecondDate : TDateTime) : Boolean;

Function ReturnBoundaryWeeksBetweenDates(StartingDayOfWeek : Integer; FirstDate, SecondDate : TDateTime) : Integer;
Function ReturnBoundaryMonthsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;
Function ReturnBoundaryQuartersBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;
Function ReturnBoundaryYearsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Function ReturnDaysBetweenDates(FirstDate, SecondDate : TDateTime) : LongInt;
Function ReturnBusinessDaysBetweenDates(FirstDate, LastDate : TDatetime) : Integer;
Function ReturnWeeksBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;
Function ReturnMonthsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;
Function ReturnQuartersBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;
Function ReturnYearsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Function IsValidDate(WorkDate : String) : Boolean;
Function IsLeapYear(WorkDate : TDateTime) : Boolean;

Function IsLeapYeari(aYear : Integer) : Boolean;
Function IsDateInRange(WorkDate, StartDate, EndDate : TDateTime) : Boolean;

Function AddDays(NumberOfDays : Integer; WorkDate :TDateTime) : TDateTime;
*Function AddBusinessDays(NumberOfDays : Integer; WorkDate :TDateTime) : TDateTime;
Function AddWeeks(NumberOfWeeks : Integer; WorkDate :TDateTime) : TDateTime;
Function AddMonths(NumberOfMonths : Integer; WorkDate :TDateTime) : TDateTime;
Function AddQuarters(NumberOfQuarters : Integer; WorkDate :TDateTime) : TDateTime;
Function AddYears(NumberOfYears : Integer; WorkDate :TDateTime) : TDateTime;

Function VerbalDate(WorkDate : TDatetime) : String;
Function FormatDate(FormatMask : String; WorkDate : TDateTime) : String;

Function DaysInMonth(WorkDate : TDateTime) : Word;
Function DaysInMonthi(Month, Year : Word) : Word;
Function DaysInYear(WorkDate : TDateTime) : Word;
Function DaysInYeari(Year : Integer) : Word;
Function DaysLeftInMonth(WorkDate : TDateTime) : Word;
Function DaysLeftInMonthi(Day, Month, Year : Integer) : Word;
Function DaysLeftInYear(WorkDate : TDateTime) :
Function DaysLeftInYeari(Day, Month, Year : Integer) : Word;

Function AddDate(DateMask : String; NumberOfPeriods : Integer; WorkDate : TDateTime) : TDateTime;
Function DateDiff(DateMask : String; StartingDayOfWeek : Integer; FirstDate, SecondDate : TDateTime) : Longint;

Function ReturnPersonsAge(DOB : TDateTime) : Integer;

Note: The 5 functions below are used by the ReturnXXXAgo functions, and the return string does not really reflect the past dates if one of the dates is not the current date.

Function ReturnDaysInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;
Function ReturnWeeksInthePast(StartingDayOfWeek : Integer; WorkDate : TDateTime; TestDate : TDateTime) : String;
Function ReturnMonthsInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;
Function ReturnQuartersInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;
Function ReturnYearsInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;

Function ReturnFirstOfTheWeek : TDateTime;
Function ReturnLastOfTheWeek : TDateTime;

Function ReturnFirstOfTheMonth : TDateTime;
Function ReturnLastOfTheMonth : TDateTime;

Function ReturnFirstOfTheYear : TDateTime;
Function ReturnLastOfTheYear : TDateTime;

Function ReturnFirstOfLastWeek : TDateTime;
Function ReturnLastOfLastWeek : TDateTime;

```
Function   ReturnFirstOfLastMonth : TDateTime;
Function   ReturnLastOfLastMonth : TDateTime;

Function   ReturnFirstOfLastYear : TDateTime;
Function   ReturnLastOfLastYear : TDateTime;

Function   IntToDate(WorkDate : Longint) : TDateTime;
Function   DateToInt(WorkDate : TDateTime) : Longint;

Function    IsHoliday(WorkDate : TDateTime) : Boolean;
Procedure SetHolidayStringList(NewHolidayStringList : TStrings);

Function    IsBusinessDay(WorkDate : TDateTime) : Boolean;
Function    IsBusinessORHoliday(WorkDate : TDateTime) : Boolean;

Function    NextBusinessDay(RequiredDate : TDateTime)   : TDateTime;
Function    PrevBusinessDay(RequiredDate : TDateTime)   : TDateTime;

Function   IsDateWithinRange(TestDate, FirstDate, LastDate : TDatetime)      : Boolean;
Function   IsDateInPosNegRange(WorkDate, CompareDate : TDateTime; PositiveDays, NegativeDays :
Integer) : Boolean;
Function   IsDateInPosNegBusinessRange(WorkDate, CompareDate : TDateTime; PositiveDays,
NegativeDays : Integer) : Boolean;
```

ReturnDelphiDayOfWeek

Declaration

Function ReturnDelphiDayOfWeek(StartingDayOfWeek, WeekDay : Integer) : Integer;

Parameters

StartingDayOfWeek : Integer, 1=Sunday, 2=Monday, etc...
WeekDay : Integer, the return value from ReturnDayOfWeek.

Description

The function ReturnDayOfWeek will return the day of the week based on any starting day for that week, 1=Sunday, 2=Monday, etc... The return value from that function is dependant on the starting day of the week. Eg: if the starting day of the week is2 (Monday) and the function returns 1 that means that the day of the week is a Monday. This function can take that return value (1) and convert it to the standard Delphi weekday numbering system, a 2, since 2 is a Monday using the standard Delphi week day numbers.

Example

```
Uses AdDates;

Var
  DayofWeek : Integer
  ADate : TAdrockDates;
Begin
   ADate := TAdrockDates.Create;
   DayOfWeek := ADate.ReturnDayOfWeek(2, Date);                    {Monday is the starting
day of the week }
   DelphiDayOfWeek = ADate.ReturnDelphiDayOfWeek(2, DayOfWeek);          { Convert
return value to standard
 Delphi Dayofweek }
   ADate.Free;
End;
```

If the date in the above example was Monday, 20th November 1995, the ReturnDayOfWeek function would return a 1 since the 2 in the ReturnDayOfWeek function means that the week starts on a monday and the return value of 1 means that it is the first day of the week, ie a Monday.

If we then use the ReturnDelphiDayOfWeek function with that return day it will translate it to the Delphi Day of Week. The return value will then be 2 since delphi uses 1=Sunday, 2=Monday, 3=Tuesday, etc..

See Also : ReturnDayofWeek

ReturnWeekofMonth

Declaration

Function ReturnWeekofMonth(StartingDayOfWeek : Integer; WorkDate : TDateTime) : Word;

Parameters

StartingDayOfWeek : 1=Sunday, 2=Monday, etc...
WorkDate : TDateTime

Description

Return the week number of the month for the date specified in WorkDate using the specified starting day of the week.

Example

If today's date is 24th November 1995 and the starting day of the week is a 2 (Monday) then the week number of the month is 4.

See Also : ReturnDayOfWeek, ReturnWeekOfYear

DaysInMonthi

Declaration

Function DaysInMonthi(Month, Year : Word) : Word;

Parameters

Month : Word
Year : Word

Description

Return the number of days in the month specified by Month, and Year.

Example

If Month = 12 (December) and Year = 1995, then the return value will be 31.

See Also : DaysInMonth

ReturnWeekDayName

Declaration

Function ReturnWeekDayName(StartingDayOfWeek, WeekDay : Integer) : String;

Parameters

StartingDayOfWeek : 1=Sunday, 2=Monday, etc...
WorkDate : TDateTime

Description

Return the week day name for the date specified in WorkDate using the specified starting day of the week.

Example

If today's date is 24th November 1995 and the starting day of the week is a 2 (Monday) then the week day name is Friday.

See Also : ReturnDelphiDayOfWeek, ReturnDayOfWeek, ReturnWeekofMonth, ReturnWeekOfYear

ReturnWeekofYear

Declaration

Function ReturnWeekofYear(StartingDayOfWeek : Integer; WorkDate : TDateTime) : Word;

Parameters

StartingDayOfWeek : 1=Sunday, 2=Monday, etc...
WorkDate : TDateTime

Description

Return the week number of the year for the date specified in WorkDate using the specified starting day of the week.

Example

If today's date is 24th November 1995 and the starting day of the week is a 2 (Monday) then the week number of the year is 328.

See Also : ReturnDayOfWeek, ReturnWeekOfMonth

ReturnDayOfWeek

Declaration

Function ReturnDayOfWeek(StartingDayOfWeek : Integer; WorkDate : TDateTime) : Word;

Parameters

StartingDayOfWeek : 1=Sunday, 2=Monday, etc...
WorkDate : TDateTime

Description

Return the starting day of the week for a date using any starting day...

Example

If today's date is 1st december 1995, then today is Friday. If we want our week to start on a Monday then this function will return the day of the week starting from 1 for today using Monday as a starting day.

```
Uses AdDates;

Var
   DayofWeek : Integer
   ADate : TAdrockDates;
Begin
    ADate := TAdrockDates.Create;
    DayOfWeek := ADate.ReturnDayOfWeek(2, Date);
    ADate.Free;
End;
```

DayOfWeek will contain 5 since Monday was the first day of the week , and Monday would have been 27th November 1995.

Example

If the StartingDay of the week was 1 which is Sunday then DayOfWeek will contain 6 since Sunday being the first day of the week , would have been 26th November 1995.

AreDatesInSameMonth

Declaration

Function AreDatesInSameMonth(FirstDate, SecondDate : TDateTime) : Boolean;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return TRUE if the dates are in the same month, but they do not have to be in the same year.

Example

If FirstDate is 1st december 1995, and SecondDate is 2nd December 1990 then AreDatesInSameMonth will return TRUE.

See Also : AreDatesInSameYear, AreDatesInSameMonthAndYear

ReturnDayOfMonth

Declaration

Function ReturnDayOfMonth(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the day of the month from a date. 1-31...

Example

If WorkDate is 1st december 1995, then the day of the month is 1.

```
Uses AdDate;

Var
  DayofMonth : Integer
  ADate      : TAdrockDates;
Begin
    ADate := TAdrockDates.Create;
    DayOfMonth := ADate.ReturnDayOfMonth(Date)
    ADate.Free;
End;
```

ReturnDayOfYear

Declaration

Function ReturnDayOfYear(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the day of the year from a date. 1-366.

Example

If WorkDate is 1st december 1995, then the day of the year is 335

ReturnYear

Declaration

```
Function    ReturnYear(WorkDate : TDateTime)    : Word;
```

Parameters

```
WorkDate                    : TDateTime
```

Description

Return the year from a date.

Example

If WorkDate is 1st december 1995, then the year is 1995.

ReturnMonth

Declaration

Function ReturnMonth(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the month from a date, 1-12.

Example

If WorkDate is 1st december 1995, then the month is 12.

ReturnQuarter

Declaration

Function ReturnQuarter(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the quarter from a date, 1-4

1st January - 31st March	= 1
1st April - 31st June	= 2
1st July - 31st September	= 3
1st October - 31st December	= 4

Example

If WorkDate is 1st december 1995, then the quarter is 4

ReturnDayTh

Declaration

Function ReturnDayTh(WorkDate : TDateTime) : String;

Parameters

WorkDate : TDateTime

Description

Return the dayth from a date, st, nd, rd, th

1, 21, 31	= st
2, 22	= nd
3, 23	= rd
the rest	= th

Example

If WorkDate is 1st december 1995, then the DayTh is 'st' for the 1st.

See Also : ReturnDayThi

ReturnDayThi

Declaration

Function ReturnDayThi(WorkDay : Integer) : String;

Parameters

WorkDay : Integer

Description

Return the dayth from a date, st, nd, rd, th

1, 21, 31	= st
2, 22	= nd
3, 23	= rd
the rest	= th

Example

If WorkDay is 5 then DayThi will return 'th' for the 5th.

See Also : ReturnDayTh

ReturnCentury

Declaration

Function ReturnCentury(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the century part of a date..

Example

If WorkDate is 1st december 1995, then ReturnCentury will return 19.

IsValidDate

Declaration

Function IsValidDate(WorkDate : String) : Boolean;

Parameters

WorkDate : String

Description

Return TRUE if the date that is in WorkDate is a valid date, ie: it can be converted to a TDateTime variable without errors.

Example

If WorkDate = '99/10/99' then IsValidDate will return FALSE

AreDatesInSameYear

Declaration

Function AreDatesInSameYear(FirstDate, SecondDate : TDateTime) : Boolean;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return TRUE if the dates are in the same year.

Example

If FirstDate is 1st december 1995, and SecondDate is 2nd December 1990 then AreDatesInSameMonth will return FALSE

See Also : AreDatesInSameMonth, AreDatesInSameMonthAndYear

AreDatesInSameMonthAndYear

Declaration

Function AreDatesInSameMonthAndYear(FirstDate, SecondDate : TDateTime) : Boolean;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return TRUE if the dates are in the same month, and in the same year.

Example

If FirstDate is 1st december 1995, and SecondDate is 2nd December 1990 then AreDatesInSameMonthAndYear will return FALSE.

See Also : AreDatesInSameMonth, AreDatesInSameYear

IsLeapyear

Declaration

Function IsLeapYear(WorkDate : TDateTime) : Boolean;

Parameters

WorkDate : TDateTime

Description

Return TRUE if the date specified in WorkDate is in a leap year.

Example

If WorkDate is 1st december 1995, then IsLeapYear will return FALSE, since 1995 is not a leap year.

See Also : IsLeapYeari

IsLeapYeari

Declaration

Function IsLeapYeari(aYear : Integer) : Boolean;

Parameters

aYear : Integer

Description

Return TRUE if aYear is a leap year.

Example

If aYear is 1995, then IsLeapYeari will return FALSE, since 1995 is not a leap year.

See Also : IsLeapYear

IsDateInRange

Declaration

Function IsDateInRange(WorkDate, StartDate, EndDate : TDateTime) : Boolean;

Parameters

StartDate : TDateTime
EndDate : TDateTime

Description

Return TRUE if the dates in within the greater than or equal to the StartDate and less than or equal to the EndDate.

Example

If WorkDate is 1st december 1995, and StartDate is 2nd December 1990 and EndDate is 20th December 1995 then IsDateInRange will return TRUE.

AddDays

Declaration

Function AddDays(NumberOfDays : Integer; WorkDate :TDateTime) : TDateTime;

Parameters

NumberOfDays : Integer
WorkDate : TDateTime

Description

Add NumberOfDays to the WorkDate and return the result. NumberOfDays can be positive or negitive.

Example

If WorkDate = 1st December 1995, and NumberOfDays = 35 then AddDays will return 5th January 1996
If WorkDate = 1st December 1995, and NumberOfDays = -35 then AddDays will return 27th October 1995.

ReturnPersonsAge

Declaration

Function ReturnPersonsAge(DOB : TDateTime) : Integer;

Parameters

DOB : TDateTime

Description

Return the age in years given a person date of birth in DOB.

Example

If DOB is 20th December 1966, and today is 1st December 1995, ReturnPersonsAge will return 28.

AddMonths

Declaration

Function AddMonths(NumberOfMonths : Integer; WorkDate :TDateTime) : TDateTime;

Parameters

NumberOfMonths : Integer
WorkDate : TDateTime

Description

Add NumberOfMonths to the WorkDate and return the result. NumberofMonths can be positive or negitive.

Example

If WorkDate = 1st December 1995, and NumberOfMonths = 35 then AddMonths will return 1st November 1998
If WorkDate = 1st December 1995, and NumberOfMonths = -35 then AddMonths will return 1st January 1993

AddYears

Declaration

Function AddYears(NumberOfYears : Integer; WorkDate :TDateTime) : TDateTime;

Parameters

NumberOfYears : Integer
WorkDate : TDateTime

Description

Add NumberOfYears to the WorkDate and return the result. NumberOfYears can be positive or negitive.

Example

If WorkDate = 1st December 1995, and NumberOfYears = 35 then AddYears will return 1st November 2030
If WorkDate = 1st December 1995, and NumberOfYears = -35 then AddYears will return 1st December 1960.

AddWeeks

Declaration

Function AddWeeks(NumberOfWeeks : Integer; WorkDate :TDateTime) : TDateTime;

Parameters

NumberOfWeeks : Integer
WorkDate : TDateTime

Description

Add NumberOfWeeks to the WorkDate and return the result. NumberOfWeeks can be positive or negitive.

Example

If WorkDate = 1st December 1995, and NumberOfMonths = 35 then AddWeeks will return 2nd August 1996
If WorkDate = 1st December 1995, and NumberOfMonths = -35 then AddWeeks will return 31st March 1995

AddQuarters

Declaration

Function AddQuarters(NumberOfQuarters : Integer; WorkDate :TDateTime) : TDateTime;

Parameters

NumberOfQuarters : Integer
WorkDate : TDateTime

Description

Add NumberOfQuarters to the WorkDate and return the result. NumberofQuarters can be positive or negitive.

Example

If WorkDate = 1st December 1995, and NumberOfQuarters = 35 then AddMonths will return 1st august 2004
If WorkDate = 1st December 1995, and NumberOfQuarters = -35 then AddMonths will return 1st February 1987

AddDate

Declaration

Function AddDate(DateMask : String; NumberOfPeriods : Integer; WorkDate : TDateTime) : TDateTime;

Parameters

DateMask	: String
NumberOfPeriods	: Integer
WorkDate	: TDateTime

Description

Add NumberOfPeriods to the WorkDate and return the result. NumberOfPeriods can be positive or negitive.
DateMask can be D = Days, W=Weeks, M=Months, Q=Quarters, Y=Years.

Example

If WorkDate = 1st December 1995, and DateMask = D and NumberOfPeriods = 35 then AddDate will return 5th Janaury 1996
If WorkDate = 1st December 1995, and DateMask = W and NumberOfPeriods = 35 then AddDate will return 2nd August 1996
If WorkDate = 1st December 1995, and DateMask = M and NumberOfPeriods = 35 then AddDate will return 1st November 1998
If WorkDate = 1st December 1995, and DateMask = Q and NumberOfPeriods = 35 then AddDate will return 1st August 2004
If WorkDate = 1st December 1995, and DateMask = Y and NumberOfPeriods = 35 then AddDate will return 1st December 2030

Note

If an invalid date mask is given the returning date will be 0

See Also : DateDiff

ReturnDaysBetweenDates

Declaration

Function ReturnDaysBetweenDates(FirstDate, SecondDate : TDateTime) : LongInt;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of days between the two the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 10th October 1995 then ReturnDaysBetweenDates will return 91

ReturnWeeksBetweenDates

Declaration

Function ReturnWeeksBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of weeks between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 10th October 1995 then ReturnWeeksBetweenDates will return 13

Note

This function returns actual weeks apart not part weeks apart.

If the first date is 1st December 1995, and the second date is the 25th of November 1995 ReturnWeeksBetweenDates will return 0 because there is not a full week between the dates.

If the first date is 1st December 1995, and the second date is the 24th of November 1995 ReturnWeeksBetweenDates will return 1 because there is a full week between the dates.

If the first date is 1st December 1995, and the second date is the 18th of November 1995 ReturnWeeksBetweenDates will return 1 because there is still only 1 full week between the dates.

See Also : ReturnBoundaryWeeksBetweenDates

ReturnBoundaryMonthsBetweenDates

Declaration

Function ReturnBoundaryMonthsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of Boundary months between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 1st January 1996 then ReturnBoundaryMonthsBetweenDates will return 1

If the first date is 18th November 1995, and the second date is the 19th of DecemberNovember 1995 ReturnBoundaryMonthsBetweenDates will return 1 because there is a month border between the dates.

Note

This function returns months seperated by borders. A month border can be just 1 day, it is when there is a change of Month.

See Also : ReturnMonthsBetweenDates

ReturnBoundaryWeeksBetweenDates

Declaration

Function ReturnBoundaryWeeksBetweenDates(StartingDayOfWeek : Integer; FirstDate, SecondDate : TDateTime) : Integer;

Parameters

StartingDayOfWeek : Integer, 1=Sunday, 2=Monday, etc..
FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of Boundary weeks between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 10th October 1995 then ReturnBoundaryWeeksBetweenDates will return 13

Note

This function returns weeks seperated by borders. A week border can be just 1 day, it is when there is a change of week.

If the first date is 18th november 1995, and the second date is the 19th of November 1995 ReturnBoundaryWeeksBetweenDates will return 1 because there is a week day border between the dates.

If the first date is 18th november 1995, and the second date is the 25th of November 1995 ReturnBoundaryWeeksBetweenDates will return 1 because there is a week day border between the dates.

If the first date is 18th november 1995, and the second date is the 26th of November 1995 ReturnBoundaryWeeksBetweenDates will return 2 because there are 2 BoundaryWeekDayBorders between the dates.

See Also : ReturnWeeksBetweenDates

ReturnMonthsBetweenDates

Declaration

Function ReturnMonthsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of months between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 10th October 1995 then ReturnMonthsBetweenDates will return 2

Note

This function returns actual months apart not part months apart.

If the first date is 1st December 1995, and the second date is the 25th of November 1995 ReturnMonthsBetweenDates will return 0 because there is not a full month between the dates.

If the first date is 1st December 1995, and the second date is the 2nd of November 1995 ReturnMonthsBetweenDates will return 0 because there is still not a a full month between the dates.

If the first date is 1st December 1995, and the second date is the 1st of November 1995 ReturnMonthsBetweenDates will return 1 because there is 1 full month between the dates.

See Also : ReturnBoundaryMonthsBetweenDates

ReturnBoundaryYearsBetweenDates

Declaration

Function ReturnBoundaryYearsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Parameters

FirstDate	: TDateTime
SecondDate	: TDateTime

Description

Return the number of Boundary years between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 1st January 1996 then ReturnBoundaryMonthsBetweenDates will return 1

If the first date is 18th November 1995, and the second date is the 19th of DecemberNovember 1995 ReturnBoundaryMonthsBetweenDates will return 1 because there is a month border between the dates.

Note

This function returns months seperated by borders. A month border can be just 1 day, it is when there is a change of Month.

See Also : ReturnYearsBetweenDates

ReturnYearsBetweenDates

Declaration

Function ReturnYearsBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of years between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 10th October 1995 then ReturnYearsBetweenDates will return 0

Note

This function returns actual years apart not part years apart.

If the first date is 1st December 1995, and the second date is the 25th of December 1994 ReturnYearsBetweenDates will return 0 because there is not a full year between the dates.

If the first date is 1st December 1995, and the second date is the 2nd of December 1994 ReturnYearsBetweenDates will return 0 because there is still not a a full year between the dates.

If the first date is 1st December 1995, and the second date is the 1st of Decmeber 1994 ReturnYearsBetweenDates will return 1 because there is 1 full year between the dates.

See Also : ReturnBoundaryYearsBetweenDates

DaysInMonth

Declaration

Function DaysInMonth(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the number of days in the month specified by WorkDate

Example

If WorkDate is 1st december 1995, then the DaysInMonth will be 31

See Also : DaysInMonthi

DaysInYear

Declaration

Function DaysInYear(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the number of days in the year specified by WorkDate, normally it will be 365, but in a leap year it will be 366.

Example

If WorkDate is 1st december 1995, then the DaysInYear will be 365

See Also : DaysInYeari

ReturnDaysAgo

Declaration

Function ReturnDaysAgo(WorkDate : TDateTime) : String;

Parameters

WorkDate : TDateTime

Description

Return a string that contains the number of days ago that the work date is from today's date.

If WorkDate = Today's date then return value = 'Today'
If WorkDate = Today's date -1 then return value = 'Yesterday'
If WorkDate = Today's date -2 then return value = '2 days ago.'
If WorkDate = Today's date +1 then return value = 'Tomorrow'
If WorkDate = Today's date +2 then return value = 'In 2 days.'

See also : ReturnWeeksAgo, ReturnMonthsAgo, ReturnQuartersAgo, ReturnYearsAgo

Note : All the ReturnXXXAgo functions use ReturnXXXInThePastFunctions...

DaysInYeari

Declaration

Function DaysInYeari(Year : Integer) : Word;

Parameters

Year : Integer

Description

Return the number of days in the year specified by Year, normally it will be 365, but in a leap year it will be 366.

Example

If Year is 1995 then DaysInYeari will return 365.

See Also : DaysInYear

FormatDate

Declaration

Function FormatDate(FormatMask : String; WorkDate : TDateTime) : String;

Parameters

FormatMask : String
WorkDate : TDateTime

Description

Very similar to the delphi FormatDateTime function and infact it uses it to produce most of the return string. But this method adds a new Mask character 'Z' that is used to return the DayTH of the date specified in WorkDate.

Example

```
Var
  adDates : TAdrockDates;
Begin
   ShowMessage('Today is '+AdDates.FormatDate('DZ MMM, YYYY', Date));
End
```

The example above would return 1st December 1995 if the Date was infact 1 December 1995. Notice the **st** has been inserted into the string after the day, that was the Z in the format.

See Also : VerbalDate

IntToDate

Declaration

Function IntToDate(WorkDate : Longint) : TDateTime;

Parameters

WorkDate : Longint

Description

Convert the number in WorkDate to a TDateTime variable.

Example

728628 corrosponds to the 1st December 1995.

See Also : DateToInt

DaysLeftInYear

Declaration

Function DaysLeftInYear(WorkDate : TDateTime) :

Parameters

WorkDate : TDateTime

Description

Return the number of days left in the year specified by the date in WorkDate.

Example

If WorkDate = 5th December 1995 then there is 365 days in 1995, 5th December 1995 is the 339th day of the year so the return value will be 26 as there is 26 days left in the year.

See Also : DaysLeftInYeari

DaysLeftInMonth

Declaration

Function DaysLeftInMonth(WorkDate : TDateTime) : Word;

Parameters

WorkDate : TDateTime

Description

Return the number of days left in the month from the date specified in WorkDate.

Example

If WorkDate contains the date 5th December 1995, then there is 31 days in December 1995, the return value will be 26 as there is 26 days left in the month.

See Also : DaysLeftInMonthi

DaysLeftInMonthi

Declaration

Function DaysLeftInMonthi(Day, Month, Year : Word) : Word;

Parameters

Day : Word
Month : Word
Year : Word

Description

Return the number of days left in the month specified by Day, Month, and Year.

Example

If Day = 5, Month = 12 (December) and Year = 1995, then there is 31 days in December 1995, the return value will be 26 as there is 26 days left in the month.

See Also : DaysLeftInMonth

DaysLeftInYeari

Declaration

Function DaysLeftInYeari(Day, Month, Year : Word) : Word;

Parameters

Day : Word
Month : Word
Year : Word

Description

Return the number of days left in the year specified by Day, Month, and Year.

Example

If Day = 5, Month = 12 (December) and Year = 1995, then there is 365 days in 1995, 5th December 1995 is the 339th day of the year, so the return value will be 26 as there is 26 days left in the year.

See Also : DaysLeftInYear

DateDiff

Declaration

Function DateDiff(DateMask : String; StartingDayOfWeek : Integer; FirstDate, SecondDate : TDateTime) : Longint;

Parameters

DateMask	: String
StartingDayOfWeek	: Integer, 1=Sunday, 2=Monday, etc...
FirstDate	: TDateTime
SecondDate	: TDateTime

Description

Return the difference between two dates.

DateMask can be D = Days, W=Weeks, M=Months, Q=Quarters, Y=Years, or BW=Boundary Weeks, BW=Boundary Months, BQ=Boundary Quarters, BW=BoundaryYears.

Weeks are when there is a full week difference, ie 7 days.
Months are when there is a full month difference, ie 30 or 31 days.
Years are when there is a full year difference, ie 365 or 366 days.

Boundary dates are when the date Boundary changes

Boundary Weeks are when there is a week change but it might or might not be a complete week, it could be just 1 day.
Boundary Months are when there is a month change, 31st december to the 1st January is 1 Boundary month since the month has changed.
Boundary Years are when there is a year change, 31st december to the 1st January is 1 Boundary year since the yearmonth has changed.

Example

A Boundary week change could be in the above example 12th November to the 13th November, or 12th November to the 18th November, where as a week change would be 12th November to the 19th November, or 12th November to the 25rd November.

Note

If an invalid date mask is given the return value will be 0

See Also : AddDate

DateToInt

Declaration

Function DateToInt(WorkDate : TDateTime) : Longint;

Parameters

WorkDate : TDateTime

Description

Convert the date in WorkDate to a LongInt variable.

Example

1st December 1995 corrosponds to 728628.

See Also : IntToDate

ReturnFirstOfTheWeek

Declaration

Function ReturnFirstOfTheWeek : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the first of the current week.

See Also : ReturnLastOfTheWeek

ReturnLastOfTheWeek

Declaration

Function ReturnLastOfTheWeek : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the last day of the current week.

See Also : ReturnFirstOfTheWeek

ReturnFirstOfTheMonth

Declaration

Function ReturnFirstOfTheMonth : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the first of the current Month and Year.

See Also : ReturnLastOfTheMonth

ReturnLastOfTheMonth

Declaration

Function ReturnLastOfTheMonth : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the Last day of the current Month and Year.

See Also : ReturnFirstOfTheMonth

ReturnFirstOfTheYear

Declaration

Function ReturnFirstOfTheYear : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the first of the Year. 1st January

See Also : ReturnLastOfTheYear

ReturnLastOfTheYear

Declaration

Function ReturnLastOfTheYear : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the last day in the Year. 31st December.

See Also : ReturnFirstOfTheYear

ReturnFirstOfLastWeek

ReturnLastOfLastWeek

ReturnFirstOfLastMonth

Declaration

Function ReturnFirstOfLastMonth : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the first of the previous month.

See Also : ReturnLastOfLastMonth

ReturnLastOfLastMonth

Declaration

Function ReturnLastOfLastMonth : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the last day of the previous month.

See Also : ReturnFirstOfLastMonth

ReturnFirstOfLastYear

Declaration

Function ReturnFirstOfLastYear : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the first of the previous year, 1st January.

See Also : ReturnLastOfLastYear

ReturnLastOfLastYear

Declaration

Function ReturnLastofLasYear : TDateTime;

Parameters

None.

Description

Return the date as a TDateTime for the last of the previous year, 31st December.

See Also : ReturnFirstOfLastYear

ReturnMonthsAgo

Declaration

Function ReturnMonthsAgo(WorkDate : TDateTime) : String;

Parameters

WorkDate : TDateTime

Description

Return a string that contains the number of months ago that the work date is from today's date.

If WorkDate = this month then return value = 'This Month'
If WorkDate = this month -1 then return value = 'Last Month'
If WorkDate = this month -2 then return value = '2 Months Ago'
If WorkDate = this month +1 then return value = 'Next Month'
If WorkDate = this month +2 then return value = 'In 2 Months'

See also : ReturnDaysAgo, ReturnWeeksAgo, ReturnQuartersAgo, ReturnYearsAgo

Note : All the ReturnXXXAgo functions use ReturnXXXInThePastFunctions...

ReturnYearsAgo

Declaration

Function ReturnYearsAgo(WorkDate : TDateTime) : String;

Parameters

WorkDate : TDateTime

Description

Return a string that contains the number of years ago that the work date is from today's date.

If WorkDate = this Year then return value = 'This Year'
If WorkDate = this Year -1 then return value = 'Last Year'
If WorkDate = this Year -2 then return value = '2 Years Ago'
If WorkDate = this Year +1 then return value = 'Next Year'
If WorkDate = this Year +2 then return value = 'In 2 Years'

See also : ReturnDaysAgo, ReturnWeeksAgo, ReturnMonthsAgo, ReturnQuartersAgo

Note : All the ReturnXXXAgo functions use ReturnXXXInThePastFunctions...

ReturnWeeksAgo

Declaration

Function ReturnWeeksAgo(StartingDayOfWeek : Integer; WorkDate : TDateTime) : String;

Parameters

StartingDayOfWeek : Integer, 1=Sunday, 2=Monday, etc..
WorkDate : TDateTime

Description

Return a string that contains the number of weeks ago that the work date is from today's date using the Starting Day of Week.

If WorkDate = this week then return value = 'This Week'
If WorkDate = this week -1 then return value = 'Last Week'
If WorkDate = this week -2 then return value = '2 Weeks Ago'
If WorkDate = this week +1 then return value = 'Next Week'
If WorkDate = this week +2 then return value = 'In 2 Weeks'

See also : ReturnDaysAgo, ReturnMonthsAgo, ReturnQuartersAgo, ReturnYearsAgo

ReturnQuartersAgo

Declaration

Function ReturnQuartersAgo(WorkDate : TDateTime) : String;

Parameters

WorkDate : TDateTime

Description

Return a string that contains the number of quarters ago that the work date is from today's date.

If WorkDate = this quarter then return value = 'This Quarter'
If WorkDate = this quarter -1 then return value = 'Last Quarter'
If WorkDate = this quarter -2 then return value = '2 Quarters Ago'
If WorkDate = this quarter +1 then return value = 'Next Quarter'
If WorkDate = this quarter +2 then return value = 'In 2 Quarters'

See also : ReturnDaysAgo, ReturnWeeksAgo, ReturnMonthsAgo, ReturnYearsAgo

Note : All the ReturnXXXAgo functions use ReturnXXXInThePastFunctions...o

ReturnQuartersBetweenDates

Declarationb

Function ReturnQuartersBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of quarters between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 4th december 1995, and SecondDate is 4th September 1995 then ReturnQuartersBetweenDates will return -1

Note

This function returns actual quarters apart not part quarters apart.

If the first date is 4th December 1995, and the second date is the 5th of September 1995 ReturnQuartersBetweenDates will return 0 because there is not a full quarter between the dates.

Quarters Are As Follows:

1st January - 31st March	= 1
1st April - 31st June	= 2
1st July - 31st September	= 3
1st October - 31st December	= 4

See Also : ReturnBoundaryQuartersBetweenDates

ReturnBoundaryQuartersBetweenDates

Declaration

Function ReturnBoundaryQuartersBetweenDates(FirstDate, SecondDate : TDateTime) : Integer;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of Boundary quarters between the first date and the second date. If the second date is later than the first then the result will be negitive.

Example

If FirstDate is 1st december 1995, and SecondDate is 1st January 1996 then ReturnBoundaryMonthsBetweenDates will return 1 since there is 1 Boundary quarter between the dates.

Note

This function returns quarters seperated by borders. A quarter border can be just 1 day, it is when there is a change of quarters.

Quarters Are As Follows:

1st January - 31st March = 1
1st April - 31st June = 2
1st July - 31st September = 3
1st October - 31st December = 4

See Also : ReturnQuartersBetweenDates

ReturnDaysInthePast

Declaration

Function ReturnDaysInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;

Parameters

WorkDate	: TDateTime
TestDate	: TDateTime

Description

Return a string that contains the number of days ago that the work date is from TestDate.

If WorkDate = TestDate then return value = 'Today'
If WorkDate = TestDate -1 then return value = 'Yesterday'
If WorkDate = TestDate -2 then return value = '2 days ago.'
If WorkDate = TestDate +1 then return value = 'Tomorrow'
If WorkDate = TestDate +2 then return value = 'In 2 days.'

See also : ReturnWeeksInThePast, ReturnMonthsInThePast, ReturnQuartersInThePast, ReturnYearsInThePast

ReturnWeeksInthePast

Declaration

Function ReturnWeeksInthePast(StartingDayOfWeek : Integer; WorkDate : TDateTime; TestDate : TDateTime) : String;

Parameters

WorkDate	: TDateTime
TestDate	: TDateTime

Description

Return a string that contains the number of weeks ago that the work date is from TestDate based on the StartingDayOfTheWeek

If WorkDate = the week of TestDate then return value = 'This Week'
If WorkDate = the week of TestDate -1 then return value = 'Last Week'
If WorkDate = the week of TestDate -2 then return value = '2 Weeks ago.'
If WorkDate = the week of TestDate +1 then return value = 'Next Week'
If WorkDate = the week of TestDate +2 then return value = 'In 2 Weeks.'

See also : ReturnDaysInThePast, ReturnMonthsInThePast, ReturnQuartersInThePast, ReturnYearsInThePast

ReturnQuartersInThePast

Declaration

Function ReturnQuartersInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;

Parameters

WorkDate	: TDateTime
TestDate	: TDateTime

Description

Return a string that contains the number of quarters ago that the work date is from TestDate.

If WorkDate = the Quarter of TestDate then return value = 'This Quarter'
If WorkDate = the Quarter of TestDate -1 then return value = 'Last Quarter'
If WorkDate = the Quarter of TestDate -2 then return value = '2 Quarters Ago.'
If WorkDate = the Quarter of TestDate +1 then return value = 'Next Quarter'
If WorkDate = the Quarter of TestDate +2 then return value = 'In 2 Quarters.'

See also : ReturnDaysInThePast, ReturnWeeksInThePast, ReturnMonthsInThePast, ReturnYearsInThePast

ReturnMonthsInThePast

Declaration

Function ReturnMonthsInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;

Parameters

WorkDate	: TDateTime
TestDate	: TDateTime

Description

Return a string that contains the number of months ago that the work date is from TestDate.

If WorkDate = the month of TestDate then return value = 'This Month'
If WorkDate = the month of TestDate -1 then return value = 'Last Month'
If WorkDate = the month of TestDate -2 then return value = '2 Months Ago.'
If WorkDate = the month of TestDate +1 then return value = 'Next Month'
If WorkDate = the month of TestDate +2 then return value = 'In 2 Months.'

See also : ReturnDaysInThePast, ReturnWeeksInThePast, ReturnQuartersInThePast, ReturnYearsInThePast

ReturnYearsInThePast

Declaration

Function ReturnYearsInthePast(WorkDate : TDateTime; TestDate : TDateTime) : String;

Parameters

WorkDate	: TDateTime
TestDate	: TDateTime

Description

Return a string that contains the number of years ago that the work date is from TestDate.

If WorkDate = the year of TestDate then return value = 'This Year'
If WorkDate = the year of TestDate -1 then return value = 'Last Year'
If WorkDate = the year of TestDate -2 then return value = '2 Years Ago.'
If WorkDate = the year of TestDate +1 then return value = 'Next Year'
If WorkDate = the year of TestDate +2 then return value = 'In 2 Years.'

See also : ReturnDaysInThePast, ReturnWeeksInThePast, ReturnQuartersInThePast, ReturnYearsInThePast

GetFirstofMonthWeekNumber

Declaration

Function GetFirstofMonthWeekNumber(StartingDayOfMonth : Integer; WorkDate : TDateTime) : Integer;

Parameters

StartingDayOfMonth : Integer, 1=Sunday, 2=Monday, etc...
WorkDate : TDateTime

Description

Return the week number of the year for the 1st of the month specified in WorkDate, based on the starting day of the week.

Example

If WorkDate contains the date 5th December 1995, then this function translates it to 1st december 1995 and will return the week number of the year based on that date, which is 48.

ReturnBusinessDaysBetweenDates

Declaration

Function ReturnBusinessDaysBetweenDates(FirstDate, SecondDate : TDateTime) : LongInt;

Parameters

FirstDate : TDateTime
SecondDate : TDateTime

Description

Return the number of business days that fall between the first date and the second date. If the second date is later than the first then the result will be negitive.

The business days are days that do not fall on a Saturday, Sunday or appear in the holiday list.

See Also : SetholidayStringList

AddBusinessDays

VerbalDate

Declaration

Function VerbalDate(WorkDate : TDateTime) : String;

Parameters

WorkDate : TDateTime

Description

Return a string that displays the date in plain english.

Example

If WorkDate is 1/12/1995, then the the string will result in 1st day of december 1995.

See Also : FormatDate

IsHoliday

Declaration

Function IsHoliday(WorkDate : TDateTime) : Boolean;

Parameters

WorkDate : TDateTime

Description

Returns true if the date that is passed in WorkDate falls on a holiday. Holidays are setup using SetHolidayStringList() passing a TStrings object, a list box items, or a string list....

See Also : IsBusinessDay, IsBusinessOrHoliday

SetHolidayStringList

Declaration

Function SetHolidayStringList(NewHolidayStringList : TStrings);

Parameters

NewHolidayStringList : TStrings;

Description

This routine sets up a holiday list of dates. Each line of the string list should contain a date in string format. The businessxxx routines use the holiday list to also restrict business dates to those that do not fall on Saturday, Sunday or any date in the List.

See Also : IsHoliday, IsBusinessDay, IsBusinessOrHoliday

IsBusinessDay

Declaration

Function IsBusinessDay(WorkDate : TDateTime) : Boolean;

Parameters

WorkDate : TDateTime

Description

Returns true if the date that is passed in WorkDate falls on a a business day (Monday-Friday)

See Also : IsHoliday, IsBusinessOrHoliday

IsBusinessORHoliday

Declaration

Function IsBusinessOrHoliday(WorkDate : TDateTime) : Boolean;

Parameters

WorkDate : TDateTime

Description

Returns true if the date that is passed in WorkDate falls on a business day (Monday-Friday), or on a holiday. Holidays are setup using SetHolidayStringList() passing a TStrings object, a list box items, or a string list....

See Also : IsHoliday, IsBusinessDay

NextBusinessDay

Declaration

Function NextBusinessDay(WorkDate : TDateTime) : TDateTime;

Parameters

WorkDate : TDateTime

Description

Returns the date for the next business day. A business day is normally any saturday, or sunday. The business routines will also use the holiday list is it has been setup to restrict dates.

See Also : PrevBusinessDay, SetHolidayStringList

PrevBusinessDay

Declaration

Function PrevBusinessDay(WorkDate : TDateTime) : TDateTime;

Parameters

WorkDate : TDateTime

Description

Returns the date for the previous business day. A business day is normally any saturday, or sunday. The business routines will also use the holiday list is it has been setup to restrict dates.

See Also : NextBusinessDay, SetHolidayStringList

IsDateWithinRange

Declaration

Function IsDateWithinRange(TestDate, FirstDate, LastDate : TDatetime) : Boolean;

Parameters

TestDate, FirstDate, LastDate : TDatetime;

Description

Returns TRUE if the testdate is between the first date and the last date.

IsDateInPosNegRange

Declaration

Function IsDateInPosNegRange(WorkDate, CompareDate : TDateTime; PositiveDays, NegativeDays : Integer) : Boolean;

Parameters

WorkDate, CompareDate: TDateTime;
PositiveDays, NegativeDays : Integer;

Description

Returns TRUE if the date specified in workdate is between (CompareDate-NegativeDays) and (CompareDate+PositiveDays)

See Also : IsDateInPosNegBusinessRange

IsDateInPosNegBusinessRange

Declaration

Function IsDateInPosNegBusinessRange(WorkDate, CompareDate : TDateTime; PositiveDays, NegativeDays : Integer) : Boolean;

Parameters

WorkDate, CompareDate: TDateTime;
PositiveDays, NegativeDays : Integer;

Description

Returns TRUE if the date specified in workdate is between (CompareDate-Negative Business Days) and (CompareDate+Positive Business Days)

Both PositiveDays, and Negative days are expressed as business days.

See Also : IsDateInPosNegRange