

Der Kalender

copyright 1987-2017, Harald-J. Fey, Hemmingen

Bei diesem **Kalender** handelt es sich **nicht** um ein **Programm** im "herkömmlichen Sinne", **sondern** um eine "**Komponente**", welche in jede beliebige software eingebaut werden kann. Die Komponente besteht aus einem visuellen Teil, welcher abgeschaltet werden kann, und aus einem nichtvisuellen Berechnungsteil. Auch können nur einzelne units verwendet werden, falls der volle Funktionsumfang nicht benötigt wird. Das Erscheinungsbild der "Ansteuerung" in der jeweiligen software kann also völlig unterschiedlich ausfallen.

Geben Sie mal in Ihrer software bei einer "Datumsabfrage" z.B. den "20.03.1598" ein; versuchen die Arbeitstage / Feiertage im Mai 2020 in Bayern zu ermitteln, eine Frist oder Zinsen etc. zu berechnen...

download

Bedienung

Kalender-Komponente

Technik

dieses Dokument als pdf laden

Kulturgeschichte

Bedienung des visuellen Teils der Komponente:

Die meisten Bedienungselemente sind durch Hinweise (hints) erklärt, gehen Sie mit dem cursor auf das entsprechende Element und es erscheint der entsprechende Hinweis.

Hier die grundlegenden Befehle:



rechte Maustaste auf Tag/Monat/Jahr =
Übernahme des Datums und des Landes

DoubleClick auf Tag/Monat/Jahr =
Übernahme des Datums und des Landes auf das Kalenderblatt
und dieses öffnen

Button ganz rechts oben = Umschaltung:
Gregorianischer - Julianischer Kalender
und umgekehrt

Button ganz rechts unten = Umschaltung

- 1.) a) **Click auf Liste:**
Feiertags- und Länderauswahl
- b) **DblClick auf Liste:**
 betreffender Feiertag
 = <http://de.wikipedia.org/wiki/> Feiertag
- 2.) Tag des Jahres (DDD.YYYY)
- 3.) Kalenderwoche (WW.YYYY)
- 4.) Julianischen Datum (JJJJJJJ)
- 5.) Datum (DD.MM.YYYY)
- 6.) = 1.)



- Kalenderblatt** = Weberschaltung / Umschaltung
- Land:** (z.B. Niedersachsen)
 Click = <http://de.wikipedia.org/wiki/> Land (Artikel)
 rechte Maustaste = <http://maps.google.com/>
- Landkarte (Landeshauptstadt)
Monat: Click = <http://de.wikipedia.org/wiki/> Monat
Jahr: Click = <http://de.wikipedia.org/wiki/> Jahr
Wochentag: Click = <http://de.wikipedia.org/wiki/> Wochentag
K-woche: Click = <http://de.wikipedia.org/wiki/woche>
Datum:
 Click = Umschaltung Datumsfenster auf diesen Tag
 DblClick = <http://de.wikipedia.org/wiki/> Datum ohne Jahr
 rechte Maustaste = Wetter,
 Sonne,
 Mond,
 Sterne
 in der Landeshauptstadt des ausgewählten Bundeslandes und
 Tide in Cuxhaven
 via <http://www.wolframalpha.com>
- Button Progr.:** = weitere Funktionen des visuellen Teils der Komponente:
Uhrzeit,
Fristenberechnung nach BGB,
Zeitdifferenzberechnung (Arbeitsstunden),
Zeitdifferenzberechnung,
Windows Info,
WKN nach ISIN / Kurs abfragen
Datumsformate (numerisch)
Datumsformate (alphanumerisch)
Datumsformate (IT)
- Button K:** = zurück zum Kalenderblatt
- Button C:** = Anzeige als Text in die Zwischenablage kopieren,

zur Verwendung in jeder beliebigen Software
(Textverarbeitung etc.)

Button S:

Click = [Suchmaschine der Leibniz Universität Hannover](#)
rechte Maustaste = Auswahl diverser Suchmaschinen, Zeitungen,

Nachrichten etc.

Uhrzeit



Fristenberechnung nach § 187 I BGB



Zeitdifferenz (Arbeitsstunden)

Zeitdifferenz (Arbeitsstunden)

09-05-2010 Niedersachsen

C K Progr.

08:00 Uhrzeit Beginn

17:32 Uhrzeit Ende

09:32 Anzahl der Stunden

Sekunden ausblenden

copyright 1987-2010, Harald-J. Fey, Hemmingen

Zeitdifferenz

Zeitdifferenz

30-04-1600 bis 08-05-2010

inklusive exklusiv

C K Progr.

Vom 30.04.1600 (inkl.) bis 08.05.2010 (exkl.):

- 410 Zeitjahre, 8 Tage
- 410 Zeitjahre, 0 Zeitmonate, 8 Tage
- 410 Zeitjahre, 0 Zeitmonate, 1 Zeitwoche, 1 Tag
- 4920 Zeitmonate, 8 Tage
- 4920 Zeitmonate, 1 Zeitwoche, 1 Tag
- 21393 Zeitwochen, 6 Tage
- 149757 Tage
- 409 Kalenderjahre, 373 Tage
(246 Tage in 1600*, 127 Tage in 2010)
- 409 Kalenderjahre, 12 Kalendermonate, 8 Tage
(8 Kalendermonate, 1 Tag in 1600*,
4 Kalendermonate, 7 Tage in 2010)

copyright 1987-2010, Harald-J. Fey, Hemmingen

Windows Info



WKN nach ISIN / Kurs abfragen



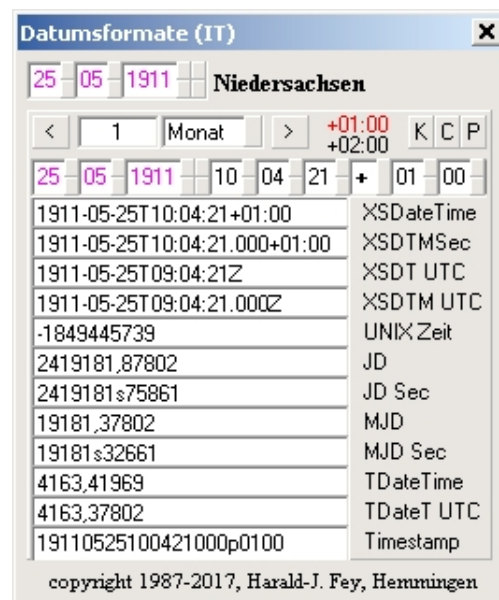
Datumsformate (numerisch)



Datumsformate (alphanumerisch)



Datumsformate (IT)



Die Kalender-Komponente

Die Delphi-Komponente Kalender beinhaltet einen [visuellen Kalender](#) (Button C und die Daten werden als txt in die Zwischenablage kopiert) und über **400 functions** zur [Datum](#) und [Zeit \(mit allen Sommerzeiten in Deutschland und Österreich\)](#), [UNIX-Time](#) (ohne Jahr-2038-Problem, ohne Jahr-1901-Problem), [XS-Time-Format](#), Zonenzeit nach UTC etc., [Julianisches Datum](#), [Modifiziertes Julianisches Datum](#), [Zeitintervall](#)-, [Zins](#)-, [Prüfzifferberechnung](#), [Rundung](#), [Text-Konvertierung](#), sowie [geographische](#) und [astronomische](#) Berechnungen; und Berechnungen zum deutschen [Steuer](#)- und [Justizkostenrecht](#) etc... Und auch einige allgemeine [windows-spezifische functions](#).

Der Kalender umfaßt den Zeitraum vom 03.01.0001 bis 31.12.2999 des [Julianischen](#) Kalenders ([alter Stil](#)) sowie das [Gregorianische](#) ([neuer Stil](#)) bzw. Vor-Gregorianische Äquivalent, die Osterfestberechnung ([gregorianisch](#) 1583 bis 2299, [julianisch](#) über den gesamten Zeitraum), der [Kalenderwoche](#) (Anzahl der Werkzeuge auch ohne Samstag) und des Tages des jeweiligen Jahres, die Feiertagsberechnung für alle [deutschen Bundesländer](#) und [österreichischen Bundesländer](#), die Berechnung der Mondphasen, des Sonnenauf- und unterganges, der Dämmerung, der [mittleren](#) und [wahren](#) Ortszeit und vieles andere mehr.

Die Zinsberechnung beinhaltet - basierend auf dem Kalender - elf unterschiedliche Zinsberechnungsmethoden:

- actual/actual im Kalenderjahr
- actual/actual im Röm. Kalenderjahr
- actual/365 im Kalenderjahr
- actual/365 im Röm. Kalenderjahr
- actual/360 im Kalenderjahr
- actual/360 im Röm. Kalenderjahr
- 30/360 im Kalenderjahr (mit 31.)
- 30/360 im Röm. Kalenderjahr (mit 31.)
- 30/360 im Zeitjahr (mit 31.)
- 30/360 im Kalenderjahr
- 30/360 im Röm. Kalenderjahr

mit jeweils wahlfreier Verzinsung des Anfangs- und/oder Enddatums sowie die Berechnung des Valutatages (Werktag und/oder Samstag etc., Fristenberechnung) für alle deutschen Bundesländer; es können unterschiedliche Rundungsmethoden gewählt und die Dezimalstellen getrennt nach interner und

externer Rundung gesetzt werden.

Es können - basierend auf dem Kalender - Zeitintervallberechnungen (Zeitjahre, -monate, -wochen, Kalenderjahre, -monate, -wochen, Tage, Stunden, Minuten, Sekunden etc.) durchgeführt werden.

Die Komponente läuft ab [Delphi 1](#) ([Windows 3.11](#), 486er, 66 MHz) und ist trotz ihres Funktionsumfangs sehr klein und sehr schnell.

Die grundlegenden functions wurden von mir Anfang der 80er Jahre auf einer [IBM 5110](#) ([IBM System/360](#), [Emulator 5110](#)) entwickelt und sind seitdem auf den unterschiedlichsten Systemen unter den verschiedensten Sprachen im Einsatz. Diese functions werden seitdem von mir ständig weiter entwickelt und neue functions kommen laufend hinzu, um zusätzliche Anforderungen erfüllen zu können.

Die Programmierung der functions ist bewußt allgemein gehalten, so daß diese weder sprach- noch systemabhängig sind. Sämtliche Kalender- und Tabellendaten etc. werden bei jedem Funktionsaufruf neu errechnet; es sind also keine diesbezüglichen "eingetippten Tabellen" vorhanden, welche nur Speicher "fressen" würden und obendrein sehr (schreib)fehleranfällig wären.

Ein umfangreiches Zinsberechnungsprogramm ([zins.exe](#) als Beispiel), welches auf dieser Komponente basiert, läßt sich unter Delphi schon mit wenigen Programmierzeilen realisieren; nur die Grafiksteuerung muß (und sollte auch!) jeder nach seinem Gusto selbst programmieren.

Sämtliche Kalenderberechnungen basieren auf dem [Julianischen Datum](#) nach [Joseph Justus Scaliger](#). Das [Julianische Datum](#) ist eine schlichte fortlaufende Zählung der Tage seit dem mittleren Mittag des 01.01.4713 v.Chr.; einfach genial diese Idee des Herrn [Scaliger](#)!

Es werden deshalb bei der gesamten Berechnung keine speziellen functions irgendeines Betriebssystems und/oder irgendeiner Programmiersprache verwendet; nur das aktuelle Datum und die Zeit wird in Pascal über date bzw. time abgefragt, mehr nicht.

Die Rundungs-functions basieren auf Text- und Integer-Werten.

Aus diesem Grunde ist das gesamte Berechnungssystem völlig autark und nicht abhängig von einem Betriebssystem und/oder einer Sprache und/oder dessen bzw. deren Genauigkeit.

Das **Julianische Datum** darf aber **KEINESFALLS** mit dem **Datum des Julianischen Kalender** verwechselt werden!

Das [Julianische Datum](#) ist NUR eine Zahl, die keine Assoziationen erwecken kann. Das Datum des [Julianischen](#) Kalenders gibt aber den Tag, den Monat und das Jahr an; ebenso wie das Datum des [Gregorianischen](#) Kalenders, der heute verwendet wird.

Der [Julianische](#) Kalender wurde am 1. Januar 45 v. Chr. von [Julius Caesar](#) eingeführt, dieser Kalender hat von der Grundstruktur her noch heute Gültigkeit.

Im Jahre 325 nahm das [Nicaenische Konzil](#) den [Julianischen](#) Kalender für die gesamte Christenheit an.

Nach der Kalenderreform des [Augustus](#) im Jahre 8 v. Chr. und der Reform durch Papst [Gregor XIII.](#) (Ostern wanderte immer weiter gen Sommer) gilt dieser Kalender nunmehr seit dem 15. Oktober 1582 unverändert. Die Gregorianische Reform fand jedoch nicht sogleich allgemeine Zustimmung, so daß der [Gregorianische](#) Kalender in den [einzelnen Ländern](#), zu unterschiedlichen Zeiten eingeführt wurde; so z.B. in Griechenland erst am 23. März 1924.

Bei historischen Daten ist deshalb Vorsicht geboten und zunächst zu prüfen, ob diese das Datum nach dem [Julianischen](#) Kalender ([alter Stil](#)) oder dem [Gregorianischen](#) Kalender ([neuer Stil](#)) nennen!

Die links zu wikipedia-Artikeln (und eventuell dort genannten Programmierbeispielen) besagen **NICHT**, daß die Kalender-Komponente so - wie dort ggf. beschrieben - programmiert ist bzw. die dort angeführten historischen, juristischen, technischen etc. Aussagen die Grundlagen sind.

Diese links zu den entsprechenden wikipedia-Artikeln dienen lediglich der Information über das äußerst komplexe Thema des Kalenders.

Die grundlegenden functions dieser [Komponente](#) wurden angeregt durch die hervorragende wissenschaftliche Abhandlung des **Semjon Issakowitsch Seleschnikow** (bis zu seinem Tode im Jahr 1970 **Mitglied des Astronomischen Rates der Akademie der Wissenschaften der UdSSR**); Moskau 1977, Köln 1981.

Technik

Für die Übergabe an die functions und für das visuelle Erscheinungsbild sind lediglich folgende Konventionen zu beachten:

Das Datum wird immer als string im Format DD.MM.YYYY übergeben.

Die Zeit wird immer als string im Format HH:MM:SS übergeben.

Der Dezimaltrenner ist immer der Decimalseparator (also entweder Komma oder Punkt, je nach der Ländereinstellung von windows).

Greg = Datum im Gregorianischen Kalender (string = DD.MM.YYYY)
Jul = Datum im Julianischen Kalender (string = DD.MM.YYYY)
Scal = Julianisches Datum (string bzw. LongInt)
Sc = Julianisches Datum (string bzw. LongInt)
JD = Julianisches Datum (string bzw. LongInt)
Rev = Datum im Gregorianischen bzw. Julianischen Kalender rückwärts geschrieben (string = YYYYMMDD)

Calendar (property) = visuelles Erscheinungsbild und Kalenderumschaltung der Komponente:

J = Julianischer Kalender (gelb hinterlegt) oder
G = Gregorianischer Kalender (weiß bzw. grau hinterlegt in der Vor-Gregorianischen Zeit)

Modus (property) = visuelles Erscheinungsbild der Komponente:

D = Datum (DD.MM.YYYY)
A = Tages des Jahres (DDD.YYYY)
W = Woche des Jahres (WW.YYYY)
J = Julianisches Datum (JJJJJJJ)
T = Time (HH:MM:SS)
TM = Time (HH:MM)

Land (property) = Ländereinstellung der Komponente im amtlichen Kurzformat z.B.:
D-NI = Niedersachsen

Land = Länderbezeichnung

Short = Länderbezeichnung im amtlichen Kurzformat z.B.:
D-NI = Niedersachsen
Long = Länderbezeichnung ausgeschrieben z.B.:
Niedersachsen = D-NI

Kurzformat:

D = Deutschland
D-BW = Baden-Württemberg
D-BY = Bayern
D-BE = Berlin
D-WB = Berlin (West)
D-OB = Berlin (Ost)
D-BB = Brandenburg
D-HB = Bremen
D-HH = Hamburg
D-HE = Hessen
D-MV = Mecklenburg-Vorpommern
D-NI = Niedersachsen
D-NW = Nordrhein-Westfalen
D-RP = Rheinland-Pfalz
D-SL = Saarland
D-SN = Sachsen
D-ST = Sachsen-Anhalt
D-SH = Schleswig-Holstein
D-TH = Thüringen

A = Österreich
A-B = Burgenland
A-K = Kärnten
A-NÖ = Niederösterreich
A-OÖ = Oberösterreich
A-S = Salzburg
A-ST = Steiermark
A-T = Tirol
A-V = Vorarlberg
A-W = Wien

sStandard in RoundExt = Rundungsverfahren:

nil = keine Rundung
german = kaufmännische Rundung
math = mathematische Rundung

iRoundModus in RoundMod = Rundungsverfahren:

-1 = keine Rundung
0 = kaufmännische Rundung
2 = mathematische Rundung

sCurrName in RatesToEUR und in CurrToEUR =
mit den festen Wechselkursen hinterlegte Währungen:

BEF, DEM, GRD, ESP, FRF, IEP, ITL, LUF, NLG, ATS, PTE, FIM

sTab in JKR = Gebührenart:

RA = Rechtsanwalt
RAP = Rechtsanwalt (Prozesskostenhilfe)
GK = Gerichtskosten
KO = Kostenordnung
GV = Gerichtsvollzieher
HEB = Hebegebühr

Klasse in EStG = Lohn- bzw. Einkommensteuertabellen:

G = EStG-Grundtabelle
S = EStG-Splittingtabelle

Lohnsteuertabellen:

1 = Steuerklasse I
2 = Steuerklasse II
3 = Steuerklasse III
4 = Steuerklasse IV
5 = Steuerklasse V
6 = Steuerklasse VI

Zeitr in EStG = Abrechnungszeitraum:

T = tägliche Lohnsteuer
W = wöchentliche Lohnsteuer
M = monatliche Lohnsteuer
J = jährliche Lohnsteuer

Der result-value der functions (bei result : integer) ist im Falle der erfolgreichen Verarbeitung = +1 (alle anderen Werte sind Fehlermeldungen).

SetParentWin(NewParent : TWinControl)

immer in OnActivate bzw. OnCreate auf das Formular setzen (damit kann auch während der Laufzeit das parentwindow gewechselt werden) und die Komponente zum Schluß freigeben; z.B.:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Kalender1.SetParentWin(self);
end;
```

```
procedure TForm1.FormDestroy(Sender: TObject);
begin
    Kalender1.Free;
end;
```

Die speziellen properties der Komponente sind folgendermaßen deklariert:

```
property Modus : string           read GetModus      write SetModus;
property Visible : boolean        read GetVisible    write SetVisible;
property ShowBtn : boolean        read GetBtn        write SetBtn;
property ShowCal : boolean        read GetShowCal    write SetShowCal;
property ShowTour : string        read GetTour        write SetTour;
property Calendar : string       read GetCalendar   write SetCalendar;
property Land : string            read GetLand        write SetLand;

property OnChange : TNotifyEvent  read FOnChange     write FOnChange;
property OnChangeVis : TNotifyEvent read FOnChangeVis  write FOnChangeVis;

property About : string           read About;
property Date : string           read GetDate;
property Day : string             read GetDay;
property Week : string            read GetWeek;
property JD : string              read GetJD;
property iJD : LongInt            read GetiJD;
property Time : string            read GetTime;
property iLeapYear : integer       read GetiLeapYear;
property iWeekDay : integer        read GetiWeekDay;
property iYearOfMoonCircle : integer read GetiYearOfMoonCircle;
property iMoonAge : integer        read GetiMoonAge;
property iMoonPhase : integer      read GetiMoonPhase;
property iHoliday : integer        read GetiHoliday;
property dtWochentag : string       read GetdtWochentag;
property Holiday : string          read Getholiday;
property HolidayInLand : string     read GetholidayInLand;
property MoonPhase : string        read GetMoonPhase;
property DateText : string        read GetDateText;
```

```

property ListDateText : TStringList read GetSlDateText;
property ListProgText : TStringList read GetSlProgText;
property CalOpen : boolean read GetCalOpen;

function SetDate(GDate : string) : boolean;
function SetDay(Day_Year : string) : boolean;
function SetWeek(Week_Year : string) : boolean;
function SetJD(JDate : string) : boolean;
function SetiJD(iJDate : LongInt) : boolean;
function SetTime(KTime : string) : boolean;

```

Alle folgenden functions etc. sind völlig unabhängig von den properties!

Die variables, procedures, functions etc. der Komponente sind folgendermaßen deklariert:

```

iScalToday : LongInt;
Today      : string;

constructor Create
    (AOwner : TComponent); override;

destructor DestroyKalend_c;

procedure SetParentWin
    (NewParent : TWinControl);

procedure OpenCal;

procedure CloseCal;

{----Scalin-functions-BEGIN ----}

function Version_Scalin : string;

function ScalToday : LongInt;

function GregToday : string;

function JulToday : string;

function ScalTest
    (ScalAsk : string;
     var iScalDate : LongInt) : integer;

function DayTest
    (DayAsk : string;
     var Day : string) : integer;

function MonthTest
    (MonthAsk : string;
     var Month : string) : integer;

function DayAndMonthTest
    (DayAsk, MonthAsk : string) : integer;

function YearTest
    (YearAsk : string;
     var Year : string) : integer;

function DayComplete
    (DayAsk : string;
     var Day : string) : integer;

function MonthComplete
    (MonthAsk : string;
     var Month : string) : integer;

```

```

function YearComplete
    (YearAsk : string;
     var Year : string) : integer;

function GregToRev
    (GregDate : string) : string;

function JulToRev
    (JulDate : string) : string;

function RevToGreg
    (sRev : string) : string;

function RevToJul
    (sRev : string) : string;

function GregToScal
    (GregDate : string;
     var ScalDate : string) : integer;

function JulToScal
    (JulDate : string;
     var ScalDate : string) : integer;

function GregToScal_L
    (GregDate, Land : string;
     var ScalDate : string;
     var iLeapYear, iDayOfTheYear, iWeekDay, iweek, iWeekOfYear, iLeapYearWeekOfYear,
     iYearOfMoonCircle, iMoonAge, iMoonPhase, iEasterSunday, iholiday,
     iFree1, iFree2, iFree3: integer;
     var dtWochentag, holiday, holidayInLand, AllHolidays, MoonPhase,
     Free1, Free2, Free3, SW_Version, SW_License, SW_Owner, SW_Valid : string) :
integer;

function JulToScal_L
    (JulDate, Land : string;
     var ScalDate : string;
     var iLeapYear, iDayOfTheYear, iWeekDay, iweek, iWeekOfYear, iLeapYearWeekOfYear,
     iYearOfMoonCircle, iMoonAge, iMoonPhase, iEasterSunday, iholiday,
     iFree1, iFree2, iFree3: integer;
     var dtWochentag, holiday, holidayInLand, AllHolidays, MoonPhase,
     Free1, Free2, Free3, SW_Version, SW_License, SW_Owner, SW_Valid : string) :
integer;

function ScalToGreg
    (ScalDate : string;
     var GregDate : string) : integer;

function ScalToJul
    (ScalDate : string;
     var JulDate : string) : integer;

function ScalToGreg_A
    (ScalDate : string;
     var GregDate : string;
     var iGregDayOfTheYear : integer) : integer;

function ScalToJul_A
    (ScalDate : string;
     var JulDate : string;
     var iJulDayOfTheYear : integer) : integer;

function ScalToGreg_L
    (ScalDate, Land : string;
     var GregDate : string;
     var iLeapYear, iDayOfTheYear, iWeekDay, iweek, iWeekOfYear, iLeapYearWeekOfYear,
     iYearOfMoonCircle, iMoonAge, iMoonPhase, iEasterSunday, iholiday,
     iFree1, iFree2, iFree3: integer;
     var dtWochentag, holiday, holidayInLand, AllHolidays, MoonPhase,

```

```

        Free1, Free2, Free3, SW_Version, SW_License, SW_Owner, SW_Valid : string) :
integer;

function ScalToJul_L
    (ScalDate, Land : string;
    var JulDate : string;
    var iLeapYear, iDayOfTheYear, iWeekDay, iweek, iWeekOfYear, iLeapYearWeekOfYear,
    iYearOfMoonCircle, iMoonAge, iMoonPhase, iEasterSunday, iholiday,
    iFree1, iFree2, iFree3: integer;
    var dtWochentag, holiday, holidayInLand, AllHolidays, MoonPhase,
    Free1, Free2, Free3, SW_Version, SW_License, SW_Owner, SW_Valid : string) :
integer;

function ScalToYear
    (iScalDate : LongInt; Calendar : integer;
    var iYear, iDayOfTheYear : LongInt; var LeapYear : boolean) : integer;

function DayAndMonthOfTheYear
    (iDayOfTheYear : integer; LeapYear : boolean;
    var iDay, iMonth : integer): integer;

function JulDayOfTheYear
    (JulDate : string) : integer;

function GregDayOfTheYear
    (GregDate : string) : integer;

function JulDayOfTheRomYear
    (JulDate : string) : integer;

function GregDayOfTheRomYear
    (GregDate : string) : integer;

function JulDayOfTheYearToJulDate
    (iDayOfTheYear, iYear : integer;
    var JulDate : string) : integer;

function GregDayOfTheYearToGregDate
    (iDayOfTheYear, iYear : integer;
    var GregDate : string) : integer;

function GregRange1582
    (iScalDate : LongInt) : integer;

function GregRange
    (iScalDate : LongInt) : integer;

function JulRange
    (iScalDate : LongInt) : integer;

function GregLeapYear
    (iYear : integer) : boolean;

function JulLeapYear
    (iYear : integer) : boolean;

function GregRomLeapYear
    (iYear : integer) : boolean;

function JulRomLeapYear
    (iYear : integer) : boolean;

function DayOfTheWeek
    (iScalDate : LongInt) : integer;

function WeekDay
    (iWeekDay : integer) : string;

function Wochentag

```

```

        (iWeekDay : integer) : string;

function JulDayOfTheWeek
    (JulDate : string) : integer;

function GregDayOfTheWeek
    (GregDate : string) : integer;

function JulWochentag
    (JulDate : string) : string;

function GregWochentag
    (GregDate : string) : string;

function ScWochentag
    (ScalDate : string) : string;

function MonthName
    (iMonth : integer) : string;

function MonatsNamen
    (iMonth : integer) : string;

function GregWeekOfTheYearToScal
    (iweek, iWeekOfYear : integer;
     var ScalDate : string) : integer;

function JulWeekOfTheYearToScal
    (iweek, iWeekOfYear : integer;
     var ScalDate : string) : integer;

function JulWeekOfTheYear
    (JulDate : string;
     var iweek, iWeekOfYear : integer ) : integer;

function GregWeekOfTheYear
    (GregDate : string;
     var iweek, iWeekOfYear : integer ) : integer;

function YearOfMoonCircle
    (iYear : integer) : integer;

function MoonAge
    (ScalDate : string;
     var MoonPhase : string) : integer;

function GregEasterSunday
    (iYear : integer) : integer;

function JulEasterSunday
    (iYear : integer) : integer;

function easter_holiday
    (iDayOfTheYear, iEasterSunday, iYear : integer;
     LeapYear : boolean; Land : string;
     var holiday, holidayInLand : string) : integer;

function calendar_holiday
    (iDayOfTheYear, iYear : integer;
     LeapYear : boolean; Land : string;
     var holiday, holidayInLand : string) : integer;

function DaysInRomMonth
    (iMonth : integer) : integer;

function DaysInMonth
    (iMonth : integer) : integer;

function GregDaysInMonth

```

```

        (iMonth, iYear : integer) : integer;

function JulDaysInMonth
    (iMonth, iYear : integer) : integer;

function GregDaysInRomMonth
    (iMonth, iYear : integer) : integer;

function JulDaysInRomMonth
    (iMonth, iYear : integer) : integer;

function GregDiffDays
    (GregDateBegin, GregDateEnd : string) : LongInt;

function JulDiffDays
    (JulDateBegin, JulDateEnd : string) : LongInt;

function DiffDays
    (iScalBegin, iScalEnd : LongInt) : LongInt;

function DiffWeeks
    (iScalBegin, iScalEnd : LongInt;
     var iTimeWeeks, iRestDaysTimeWeeks : LongInt) : integer;

function DiffCalWeeks
    (iScalBegin, iScalEnd : LongInt;
     var iCalWeeks,
     iRestDaysCalWeeks : LongInt) : integer;

function DiffMonths
    (BeginDate, EndDate : string; Calendar : integer;
     var iTimeMonths, iRestDaysTimeMonths : LongInt) : integer;

function DiffCalMonths
    (BeginDate, EndDate : string;
     var iCalMonths,
     iRestDaysCalMonths : LongInt) : integer;

function DiffMonthsYears
    (iTimeMonths : LongInt;
     var iTimeMonths1, iTimeYears : LongInt) : integer;

function DiffYears
    (BeginDate, EndDate : string; Calendar : integer;
     var iTimeYears, iRestDaysTimeYears : LongInt) : integer;

function DiffCalYears (BeginDate, EndDate : string; var iCalYears,
    iRestDaysCalYears : LongInt) : integer;

function MonthsDiffDays
    (BeginDate : string; iMonthsDiff : LongInt;
     Calendar : integer;
     var iMonthsDiffDays : LongInt) : integer;

function YearsDiffDays
    (BeginDate : string; iYearsDiff : LongInt;
     Calendar : integer;
     var iYearsDiffDays : LongInt) : integer;

function MonthsAndYearsDiffDays
    (BeginDate : string; iMonthsDiff,
     iYearsDiff : LongInt;
     Calendar : integer;
     var iMonthsAndYearsDiffDays : LongInt) : integer;

function LandToShort
    (sLandLong : string) : string;

function ShortToLand

```



```

        (sLandShort : string) : string;

function LandNames
    (iLandNr : integer) : string;

function GregCalYearMonthDayDiff
    (sDateBegin, sDateEnd : string;
     bDateBegin, bDateEnd : boolean;
     var iCalYears, iCalYearDays,
         iCalYearBeginDays, iCalYearEndDays,
         iCalYearMonths,
         iCalYearBeginMonths, iCalYearBeginMonthsDays,
         iCalYearMonthsDays,
         iCalYearEndMonths, iCalYearEndMonthsDays : integer;
     var bLeapYearBegin, bLeapYearEnd : boolean) : boolean;

function GregTimeYearMonthDayDiff
    (sDateBegin, sDateEnd : string;
     bDateBegin, bDateEnd : boolean;
     var iTimeYears,
         iTimeYearDays,
         iTimeYearMonths, iTimeYearMonthsDays : integer) : boolean;

function GregDayCorrect
    (GregDateTest : string;
     var GregDateCorr : string) : boolean;

function GregRomCalYearMonthDayDiff
    (sDateBegin, sDateEnd : string;
     bDateBegin, bDateEnd : boolean;
     var iCalYears,
         iCalYearDays,
         iCalYearBeginDays, iCalYearEndDays,
         iCalYearMonths,
         iCalYearBeginMonths, iCalYearBeginMonthsDays,
         iCalYearMonthsDays,
         iCalYearEndMonths, iCalYearEndMonthsDays : integer;
     var bLeapYearBegin, bLeapYearEnd : boolean) : boolean;

function CalToRomCal
    (sDate : string) : string;

function RomCalToCal
    (sDate : string) : string;

function GregTimeDayDiff
    (sDateBegin, sDateEnd : string;
     bDateBegin, bDateEnd : boolean;
     var iTimeDays : LongInt) : boolean;

function GregTimeWeekDayDiff
    (sDateBegin, sDateEnd : string;
     bDateBegin, bDateEnd : boolean;
     var iTimeWeeks, iTimeDays : LongInt) : boolean;

function DaysToTimeWeeks
    (iAllTimeDays : LongInt;
     var iTimeWeeks, iTimeDays : LongInt) : boolean;

function DaysToNormalLeapYear
    (iYearBeginDays, iYearEndDays : integer;
     bLeapYearBegin, bLeapYearEnd : boolean;
     var iRentNormalYearDays, iRentLeapYearDays,
         iLeapYearBegin, iLeapYearEnd : integer) : boolean;

function act_act_CalYear
    (sDateBegin, sDateEnd : string;
     bDateBegin, bDateEnd : boolean;
     var iRentYears,

```

```

        iRentNormalYearDays, iRentLeapYearDays,
        iLeapYearBegin, iLeapYearEnd : integer) : boolean;

function act_act_RomCalYear
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentNormalYearDays, iRentLeapYearDays,
        iLeapYearBegin, iLeapYearEnd : integer) : boolean;

function act_max360_CalYear
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function act_max360_RomCalYear
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function act_max365_CalYear
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function act_max365_RomCalYear
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function TimeYear_30_360
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function CalYear_30_360
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function RomCalYear_30_360
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function TimeYear_30_360_w31
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function CalYear_30_360_w31
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

function RomCalYear_30_360_w31
    (sDateBegin, sDateEnd : string;
    bDateBegin, bDateEnd : boolean;
    var iRentYears,
        iRentDays : integer) : boolean;

```

```

function ScalGregWorkDay
    (iScalDate : LongInt; sLand : string) : string;

function ScalGregWorkDayWithSa
    (iScalDate : LongInt; sLand : string) : string;

function GregWorkDay
    (GregDate, Land : string) : string;

function GregWorkDayWithSa
    (GregDate, Land : string) : string;

function GregToSc
    (GregDate : string) : string;

function JulToSc
    (JulDate : string) : string;

function ScToGreg
    (ScalDate : string) : string;

function ScToJul
    (ScalDate : string) : string;

function JulToGreg
    (JulDate : string) : string;

function GregToJul
    (GregDate : string) : string;

function Rent
    (iRentModus : integer;
     sDateBegin, sDateEnd : string;
     bDateBegin, bDateEnd : boolean;
     var iRentYears, iRentDays,
         iRentNormalYearDays, iRentLeapYearDays : integer;
     var sRentNormalYearDays, sRentLeapYearDays : string) : boolean;

function ScalGregWorkDayBefore
    (iScalDate : LongInt; sLand : string) : string;

function ScalGregWorkDayWithSaBefore
    (iScalDate : LongInt; sLand : string) : string;

function GregWorkDayBefore
    (GregDate, Land : string) : string;

function GregWorkDayWithSaBefore
    (GregDate, Land : string) : string;

function IsScalGregWorkDay
    (iScalDate : LongInt; sLand : string) : boolean;

function IsScalGregWorkDayWithSa
    (iScalDate : LongInt; sLand : string) : boolean;

function IsScalHoliday
    (iScalDate : LongInt; Land : string) : boolean;

function IsGregWorkDay
    (GregDate, Land : string) : boolean;

function IsGregWorkDayWithSa
    (GregDate, Land : string) : boolean;

function IsGregHoliday
    (GregDate, Land : string) : boolean;

```

```

function DaysOfWeek
    (GregWeek, Land : string;
     var WorkDay, WorkDayWithSa, Holiday : integer) : integer;

function DaysOfMonth
    (GregMonth, Land : string;
     var WorkDay, WorkDayWithSa, Holiday : integer) : integer;

function DaysOfYear
    (GregYear, Land : string;
     var WorkDay, WorkDayWithSa, Holiday : integer) : integer;

function ExGregDay
    (GregDate : string) : string;

function ExGregMonth
    (GregDate : string) : string;

function ExGregYear
    (GregDate : string) : string;

function PlusDay
    (GregDate : string; iDay : integer) : string;

function PlusWeek
    (GregDate : string; iWeek : integer) : string;

function PlusMonth
    (GregDate : string; iMonth : integer) : string;

function PlusYear
    (GregDate : string; iYear : integer) : string;

function MinusDay
    (GregDate : string; iDay : integer) : string;

function MinusWeek
    (GregDate : string; iWeek : integer) : string;

function MinusMonth
    (GregDate : string; iMonth : integer) : string;

function MinusYear
    (GregDate : string; iYear : integer) : string;

function PayDate
    (sTerminDate, sPayModus, sPayKat, sPayLand : string) : string;

function EquationOfTime
    (iJD : LongInt) : extended;

function MOZtoUTCDiff
    (Long : extended) : extended;

function SunDecl
    (iJD : LongInt) : extended;

function SunRiseSet
    (iJD : LongInt; Twilight : integer; Lat : extended) : extended;

function SunRise
    (iJD : LongInt; Twilight : integer; Lat, Long : extended) : extended;

function SunSet
    (iJD : LongInt; Twilight : integer; Lat, Long : extended) : extended;

function IsMESZ
    (iJD : LongInt;
     var Since : integer) : integer;

```

```

function IsMESZEx (iJD : LongInt; Land : string;
    var Since : integer;
    var TimeM, ChangeText : string) : integer;

function LastWeekDayinMonth
    (iGregMonth, iGregYear : integer; WeekDay : integer) : LongInt;

function WeekDayXinMonth
    (iGregMonth, iGregYear, iXWeekday : integer; WeekDay : integer) : LongInt;

function ShortToCapitalCity
    (sLandShort : string;
    var NorthLat, EastLong : string) : string;

function CapitalCity
    (sLandShort : string) : string;

function DateEmailGermany : string;

function DateTextGermany : string;

function LastDayInMonth
    (GregDate : string) : string;

function MonthEndBeforeYearLater
    (GregDate : string) : string;

function HolidaySpace
    (GregDate : string) : string;

function HolidaySpaceEx
    (GregDate : string) : string;

function HolidaySpaceIsIn
    (GregDateBegin, GregDateEnd : string) : string;

function HolidaySpaceIsInEx
    (GregDateBegin, GregDateEnd : string) : string;

function ScalHolidayName
    (iScalDate : LongInt; Land : string) : string;

function GregHolidayName
    (GregDate, Land : string) : string;

function ScalHolidayLegalName
    (iScalDate : LongInt; Land : string) : string;

function GregHolidayLegalName
    (GregDate, Land : string) : string;

function DateOk
    (sGDate : string) : boolean;

function DateGregOk
    (sGDate : string) : boolean;

function DateJulOk
    (sGDate : string) : boolean;

{----Scalin-functions-END ----}

{----Round-functions-BEGIN ----}

function Version_Round : string;

function RoundExt
    (eInput : extended; iDeci : integer; sStandard : string;

```

```

    var eOutput : extended) : string;

function RoundIEEE
    (eInput : extended; iDeci : integer;
    var eOutput : extended) : string;

function RoundMath
    (eInput : extended; iDeci : integer;
    var eOutput : extended) : string;

function RoundGerman
    (eInput : extended; iDeci : integer;
    var eOutput : extended) : string;

function RoundMod
    (iRoundModus : integer; eInput : extended; iDeci : integer;
    var eOutput : extended) : string;

function FillBeginZeroInt
    (input : integer; Len : integer ) : string;

function FillEndZero
    (sInput : string; Len : integer ) : string;

function FillBeginSpace
    (sInput : string; Len : integer ) : string;

function FillBeginZero
    (sInput : string; Len : integer ) : string;

function DelEndDeciZero
    (sInput : string) : string;

function DelBeginZero
    (sInput : string) : string;

function TestNum
    (sInput : string) : string;

function RentSum_30_360
    (iRoundModus, iDeciInt, iDeciExt : integer;
    eCapital, eRent : extended;
    iRentYears, iRentDays : integer;
    var sRentYear,
        sRentDay,
        sRentYearsSum,
        sRentDaysSum,
        sRentSum, sSum : string) : boolean;

function RentSum_act_act
    (iRoundModus, iDeciInt, iDeciExt : integer;
    eCapital, eRent : extended;
    iRentYears,
    iRentNormalYearDays, iRentLeapYearDays : integer;
    var sRentYear,
        sRentNormalYearDay, sRentLeapYearDay,
        sRentYearsSum,
        sRentNormalYearDaysSum, sRentLeapYearDaysSum,
        sRentSum, sSum : string) : boolean;

function RentSum_act_max360
    (iRoundModus, iDeciInt, iDeciExt : integer;
    eCapital, eRent : extended;
    iRentYears, iRentDays : integer;
    var sRentYear,
        sRentDay,
        sRentYearsSum,
        sRentDaysSum,
        sRentSum, sSum : string) : boolean;

```

```

function RentSum_act_max365
    (iRoundModus, iDeciInt, iDeciExt : integer;
    eCapital, eRent : extended;
    iRentYears, iRentDays : integer;
    var sRentYear,
        sRentDay,
        sRentYearsSum,
        sRentDaysSum,
        sRentSum, sSum : string) : boolean;

function RatesToEUR
    (sCurrName : string) : string;

function CurrToEUR
    (sCurrName : string; eCurr : extended; iDeci : integer;
    var eEUR : extended) : string;

function ExpToNormal
    (sInput : string) : string;

function Percent
    (iRoundModus, iDeciInt, iDeciExt, iDeciPer : integer;
    eCapital, ePercent : extended;
    var sPer100,
        sPer100Sum,
        sPerX,
        sPerXSum,
        sPer100_X,
        sPer100_XSum : string) : boolean;

function Perc
    (iRoundModus, iDeciExt, iDeciPer : integer;
    eCapital, ePercent : extended) : string;

function JKR
    (sDate, sTab : string; bOst : boolean;
    eWert, eZehnt : extended;
    var iOst : integer; sText : string; eGeb : extended) : string;

function EStG
    (Eink : extended; Year : integer; Klasse, Zeitr : string) : extended;

function IntG
    (eInput : extended) : extended;

function ISIN_Pruef
    (s : string) : string;

function ISIN_Input
    (s : string) : string;

function WKN_Input
    (s : string) : string;

function EuropVersReise
    (sModus, sBef, sReiseBegin : string; eWert : Extended; iPers, iTage :
integer;
    var eGeb, eGeb1Pers : Extended;
    var sGeb1Pers, sTarif, sVers : string) : string;

function PicFormat
    (iMaxWidth, iMaxHeight, iPicWidth, iPicHeight : integer;
    var iWidth, iHeight : integer) : integer;

{----Round-functions-END ----}

{----Time-functions-BEGIN ----}

```

```

function Version_Time : string;

function TimeStrToTimeInt
  (TimeIn : string;
   var iHour, iMin, iSec : LongInt) : boolean;

function TimeStrToSecInt(TimeIn : string;
  var iAllSec : LongInt) : boolean;

function SecToTimeStr
  (Sec : LongInt;
   var Days, TimeStr : string) : boolean;

function TimeTest
  (TimeIn : string;
   var TimeOut : string) : boolean;

function SetBeginZero
  (sInput : string) : string;

function TimeBeginToEnd
  (TimeBegin, TimeEnd : string;
   var TimeDiff : string) : boolean;

function TimeDiffToBegin
  (TimeDiff, TimeEnd : string;
   var TimeBegin : string) : boolean;

function TimeDiffToEnd
  (TimeDiff, TimeBegin : string;
   var TimeEnd : string) : boolean;

function TimeAdd
  (Time1, Time2 : string;
   var Days, Time3 : string) : boolean;

function DegDigitalToSec
  (DegDig : extended) : LongInt;

function DayDigitalToSec
  (Day : extended) : LongInt;

function HourDigitalToSec
  (Hour : extended) : LongInt;

function MinDigitalToSec
  (Min : extended) : LongInt;

function SecToDegDigital
  (sec : LongInt) : extended;

function SecToDayDigital
  (sec : LongInt) : extended;

function SecToHourDigital
  (sec : LongInt) : extended;

function SecToMinDigital
  (sec : LongInt) : extended;

function CoordinatesStrToDegDigital
  (Coordinates : string) : extended;

function TimeHof
  (sTimeBegin : string;
   var sTimeHof, sTimeWoerm : string) : string;

function UTCToMEZ
  (UTC : string) : string;

```



```

function UTCtoMESZ
    (UTC : string) : string;

function UTCtoOESZ
    (UTC : string) : string;

function UTCtoMSD
    (UTC : string) : string;

function MEZtoUTC
    (MEZ : string) : string;

function MESZtoUTC
    (MESZ : string) : string;

function OESZtoUTC
    (OESZ : string) : string;

function MSDtoUTC
    (MSD : string) : string;

function SecToSecDayEnd
    (iSec : integer) : integer;

function SecNowToSecDayEnd : integer;

function TimeSecNow : integer;

{----Time-functions-END ----}

{----Utils-functions-BEGIN ----}

function ComputerName : string;

function UserName : string;

function OwnerName : string;

function WinID : string;

function IsWinNT : boolean;

function WinVersInfo : string;

procedure CopyFile
    (const FileName, DestName: string);

procedure MoveFile
    (const FileName, DestName: string);

function GetFileSize
    (const FileName: string): LongInt;

function FileDateTime
    (const FileName: string): TDateTime;

function HasAttr
    (const FileName: string; Attr: Word): Boolean;

function ExecuteFile
    (const FileName, Params, DefaultDir: string; ShowCmd: Integer): THandle;

function ExecuteFileS
    (const FileName, Params, DefaultDir: string;
     ShowCmd: Integer): THandle;

function ExecuteFileDoc
    (const FileName, Params, DefaultDir: string;

```

```
        ShowCmd: Integer): THandle;

function JDtoDate
    (iJD : LongInt) : LongInt;

function DatetoJD
    (iDate : LongInt) : LongInt;

function LoadStrFromFile
    (Filename: string) : string;

function SaveStrToFile
    (Filename, SaveStr : string) : LongInt;

function ScreenPix
    (ScWidth, ScHeight : integer) : string;

function BinToChar
    (input : char) : string;

function BinToChar8
    (input : char) : string;

function Bin8ToDec
    (s : string) : integer;

function Bin8ToDecStr
    (s : string) : string;

function Bin8ToHex2
    (s : string) : string;

function Hex2ToBin8
    (s : string) : string;

function Hex2ToDec
    (s : string) : integer;

function Hex2ToDecStr
    (s : string) : string;

function DecToHex2
    (s : string) : string;

function DecToBin8
    (s : string) : string;

function BinToHex
    (input : string; columns : integer) : string;

function BinToDec
    (input : string; columns : integer) : string;

function BinToStr
    (input : string; columns : integer) : string;

function Bin8ToAlphaNum
    (sBin8 : string) : string;

function DecToAlphaNum
    (sDec : string) : string;

function HexToAlphaNum
    (sHex : string; columns : integer) : string;

function BCD_Hex1ToBCD_Bin4
    (sBCD_Hex1 : string) : string;

function BCD_Bin4ToBCD_Hex1
```

```

        (sBCD_Bin4 : string) : string;

function BCD_Hex2ToBCD_Bin8
    (sBCD_Hex2 : string) : string;

function BCD_Bin8ToBCD_Hex2
    (sBCD_Bin8 : string) : string;

function BCD_Hex1ToDec
    (sHex1 : string) : string;

function BCD_DecToHex1
    (sDec : string) : string;

function BinChecksum
    (input : string) : Char;

function GetFileVersion
    (const FileName: String): String;

function GetStackSize : integer;

function GetEnviroVar
    (const VarName: string) : string;

function ControlChar
    (iDec : integer) : string;

{----Utils-functions-END ----}

{----Text-functions BEGIN ----}

function Version : string;

function Change9To32
    (input : string) : string;

function Change10To13
    (input : string) : string;

function Change10To1310
    (input : string) : string;

function Change13To10
    (input : string) : string;

function Change13To1310
    (input : string) : string;

function Del9
    (input : string) : string;

function Del10
    (input : string) : string;

function Del13
    (input : string) : string;

function Del1013
    (input : string) : string;

function DelControls
    (input : string) : string;

function DelAllControls
    (input : string) : string;

function MACToANSI
    (input : string) : string;

```

```
function ANSIToMAC
    (input : string) : string;

function ANSIToHTML
    (input : string) : string;

function ANSIToXML
    (input : string) : string;

function ANSITo7bit
    (input : string) : string;

function ANSIToFilename
    (input : string) : string;

function DateToShortY
    (sDate : string) : string;

function DateToNotY
    (sDate : string) : string;

function DateToNot0
    (sDate : string) : string;

function DateToNot0NotY
    (sDate : string) : string;

function TimeToNot0Point
    (sTime : string) : string;

function Change00ToLine
    (sValue : string) : string;

function ReplaceStr
    (sWhole, sOldSub, sNewSub : string) : string;

function Set1000Sep
    (sValue : string) : string;

function Del1000Sep
    (sValue : string) : string;

function TurnNames
    (input : string) : string;

function TurnNamesA
    (input : string) : string;

function ANSIToURI
    (input : string) : string;

function HolidayURL
    (sHolidayName : string): string;

function LatLongToText
    (LatLong : string): string;

function ANSIToQP
    (input : string) : string;

function NumOnly
    (input : string) : string;

function Encode
    (input, code : string; modus : integer) : string;

function Decode
    (input, code : string; modus : integer) : string;
```

```

function EncodeProg
    (input : string) : string;

function ANSIToRTF
    (input : string) : string;

function PosLast
    (search, input : string) : integer;

function FirstNameComma
    (input : string) : string;

function DelTags
    (input : string) : string;

function DelTagsEx
    (input, First, Last : string) : string;

function Plural
    (iInput : integer; Seperator, Singular, PluralChar : string) : string;

function Plurale
    (iInput : integer; Seperator, Singular : string) : string;

function SortStringsWithInt
    (input : string) : string;

function SeatNrTog
    (input : string) : string;

function AdrMake
    (TermA, Name1, Name2, Name3, NameEx, Title,
     Land, Postcode, City, Street, Hs : string;
     var TermL, sign : string) : string;

function SelectHsNr
    (StreetAndHsNr : string;
     var Street, HsNr : string) : boolean;

{----Text-functions END ----}

{----common-functions BEGIN ----}

function MapURL
    (sLandShort : string): string;

function CapitalCityURL
    (sLandShort : string): string;

function LandURL
    (sLandShort : string): string;

function MapStreetURL
    (Land, PLZ, Ort, Strasse, HausNr : string) : string;

function EfaLink
    (StartCity, StartStreet : string;
     StartAdress : boolean;
     DestCity, DestStreet : string;
     DestAdress : boolean;
     sDate, sTime : string;
     StartTime : boolean) : string;

function Write_HolidayList
    (sYear, sLand : string; bGregCal : boolean): string;

{----common-functions END ----}

```

```
{----DateForm-functions BEGIN ----}
```

```
function Version_HfDateForm : string;
```

```
function MonthName_FR (iMonth : integer) : string;
```

```
function MonthName_ES (iMonth : integer) : string;
```

```
function DDMMYYYY (sGD : string) : string;
```

```
function DDtMMtYYYY (sGD : string) : string;
```

```
function DDtMMtYY (sGD : string) : string;
```

```
function DtMtYYYY (sGD : string) : string;
```

```
function DtMtYY (sGD : string) : string;
```

```
function DDsMMsYYYY (sGD : string) : string;
```

```
function DDsMMsYY (sGD : string) : string;
```

```
function DsMsYYYY (sGD : string) : string;
```

```
function DsMsYY (sGD : string) : string;
```

```
function DDpMMpYYYY (sGD : string) : string;
```

```
function DDpMMpYY (sGD : string) : string;
```

```
function DpMpYYYY (sGD : string) : string;
```

```
function DpMpYY (sGD : string) : string;
```

```
function DDpMdeYYYY (sGD : string) : string;
```

```
function DpMdeYYYY (sGD : string) : string;
```

```
function DpMdeYY (sGD : string) : string;
```

```
function DDMenYYYY (sGD : string) : string;
```

```
function DMenYYYY (sGD : string) : string;
```

```
function DMenYY (sGD : string) : string;
```

```
function DDMfrYYYY (sGD : string) : string;
```

```
function DMfrYYYY (sGD : string) : string;
```

```
function DMfrYY (sGD : string) : string;
```

```
function DDMesYYYY (sGD : string) : string;
```

```
function DMesYYYY (sGD : string) : string;
```

```
function DMesYY (sGD : string) : string;
```

```
function YYYYMMDD (sGD : string) : string;
```

```
function YYMMDD (sGD : string) : string;
```

```
function YYYYtMMtDD (sGD : string) : string;
```

```
function YYtMMtDD (sGD : string) : string;
```

```
function YYYYtMtD (sGD : string) : string;
```

```
function YYtMtD (sGD : string) : string;
```

```
function YYYYsMMsDD (sGD : string) : string;
```

```
function YYsMMsDD (sGD : string) : string;
```

```
function YYYYsMsD (sGD : string) : string;
```

```
function YYsMsD (sGD : string) : string;
```

```
function MMDDYYYY (sGD : string) : string;
```

```
function MMtDDtYYYY (sGD : string) : string;
```

```
function MMtDDtYY (sGD : string) : string;
```

```
function MtDtYYYY (sGD : string) : string;
```

```
function MtDtYY (sGD : string) : string;
```

```
function MMsDDsYYYY (sGD : string) : string;
```

```
function MMsDDsYY (sGD : string) : string;
```

```
function MsDsYYYY (sGD : string) : string;
```

```
function MsDsYY (sGD : string) : string;
```

```
function MenDDcYYYY (sGD : string) : string;
```

```
function MenDcYYYY (sGD : string) : string;
```

```
function MenDcYY (sGD : string) : string;
```

```
function ZoneDateTimeToUNIX
    (DateIn, TimeIn, DiffToUTC : string) : string;

function UNIXToUTC
    (sUNIX : string) : string;

function UNIXToUTCExt
    (sUNIX : string;
     var UTCDate, UTCTime : string) : string;

function ZoneDateTimeToUTCExt
    (DateIn, TimeIn, DiffToUTC : string;
     var UTCDate, UTCTime : string) : string;

function ZoneDateTimeToUTC
    (DateIn, TimeIn, DiffToUTC : string) : string;

function UTCToZoneDateTimeExt
    (UTCDate, UTCTime, DiffToUTC : string;
     var ZoneDate, ZoneTime : string) : string;

function UTCToZoneDateTime
    (UTCDate, UTCTime, DiffToUTC : string) : string;

function ZoneDateTimeToXSDateTime
    (ZoneDate, ZoneTime, DiffToUTC : string) : string;

function ZoneDateTimeToXSUTCDateTime
    (ZoneDate, ZoneTime, DiffToUTC : string) : string;

function XSDateTimeToZoneDateTime
    (XSDateTime : string) : string;

function XSDateTimeToZoneDateTimeExt
    (XSDateTime : string;
     var ZoneDate, ZoneTime, DiffToUTC : string) : string;

function ZoneDateTimeNowToXSUTCDateTime : string;

function ZoneDateTimeNowToXSDateTime : string;

function ZoneDateTimeNowToXSUTCDateTimeMil : string;

function ZoneDateTimeNowToXSDateTimeMil : string;

function ZoneDateTimeNowToJD : string;

function ZoneDateTimeNowToJD_Sec : string;

function ZoneDateTimeNowToJD_SecExt
    (var JD, JDsec : string) : string;

function ZoneDateTimeNowToJD_SecM : string;

function ZoneDateTimeNowToJD_SecMExt
    (var JD, JDMsec : string) : string;

function ZoneDateTimeNowToMJD : string;

function ZoneDateTimeNowToMJD_Sec : string;

function ZoneDateTimeNowToMJD_SecExt
    (var JD, JDsec : string) : string;

function ZoneDateTimeNowToMJD_SecM : string;

function ZoneDateTimeNowToMJD_SecMExt
    (var JD, JDMsec : string) : string;
```

```

function ZoneDateTimeNowToUNIX : string;

function JDToMJD (eJD : extended) : string;

function MJDTToJD (eMJD : extended) : string;

function UTCToMJD (UTCDate, UTCTime : string) : string;

function UTCToMJD_Sec (GregDate, UTCTime : string) : string;

function UTCToMJD_SecExt (GregDate, UTCTime : string;
    var MJD, Sec : string) : string;

function UTCToMJD_MSec (GregDate, UTCTimeM : string) : string;

function UTCToMJD_MSecExt (GregDate, UTCTimeM : string;
    var MJD, MSec : string) : string;

function MJDtoUTC (eMJD : extended) : string;

function MJDtoUTCExt (eMJD : extended;
    var UTCDate, UTCTime : string) : string;

function MJD_SecToUTC (MJD_Sec : string) : string;

function MJD_SecToUTCExt (MJD_Sec : string;
    var UTCDate, UTCTime : string) : string;

function MJD_MSecToUTC (MJD_MSec : string) : string;

function MJD_MSecToUTCExt (MJD_MSec : string;
    var UTCDate, UTCTime : string) : string;

function ZoneDateTimeMToYYYYMMDD_HHMMSSMMM (sDate, sSecM, sUTCDiff : string) :
string;

function ZoneDateTimeMNowToYYYYMMDD_HHMMSSMMM : string;

function ZoneDateTimeMToUTCYYYYMMDD_HHMMSSMMM (sDate, sSecM, sUTCDiff : string) :
string;

function ZoneDateTimeMNowToUTCYYYYMMDD_HHMMSSMMM : string;

function YYYYMMDD_HHMMSSMMMToZoneXSDateTimeM (YYYYMMDD_HHMMSSMMM : string) : string;

function YYYYMMDD_HHMMSSMMMToUTCXSDateTimeM (YYYYMMDD_HHMMSSMMM : string) : string;
{----DateForm-functions END ----}

```

Hinweis:

Alle Versionen dieser Komponente und/oder auf dieser basierenden Programme, die in der Versions-Nr. ein "s" beinhalten, sind **NICHT lizenzierte Versionen** und diese werden sich irgendwann "verabschieden"! Und es sind natürlich **NICHT alle functions freigeschaltet**.

**Die Nutzung ist nur für private und wissenschaftliche Zwecke zulässig.
Jede kommerzielle Nutzung OHNE Lizenz ist verboten!**

**For private and scientific use ONLY.
No commercial use without a license!**

Sólo para uso privado y científico.

No uso comercial sin licencia!

Die Nutzung erfolgt unter Ausschluß jeglicher Gewährleistung des Autors für die Richtigkeit der Ergebnisse.

download: [calendar.exe](#)

[IS Version \(calendar.exe mit Bildern\)](#)

Falls Sie diese Komponente kommerziell nutzen wollen und/oder ähnliche Software - auch datenbankgestützt - benötigen, [sprechen Sie mich einfach an...](#)

version: b71001 4.284.52.2

copyright © 1987-2017

[Harald-J. Fey](#)

Göttinger Landstr. 3

D-30966 Hemmingen

<http://haraldfey.de>