



EntwicklerCamp 2012

Notes und die Zeit

Bernhard Köhler

Vorstellung

- Quereinsteiger: Studium Physik /
Astronomie
- Assembler, Pascal, eigenes DB-Format
- 1992: Beginn der Notes-Entwicklung
- Seit 2004 selbständig in Kooperation
mit Kollegen aus Österreich und
Deutschland

Ist Ihnen so etwas schon passiert?

The image shows four overlapping error dialog boxes from IBM Lotus Notes. Each dialog has a yellow warning icon and an 'OK' button. The messages are:

- Type mismatch
- Object variable not set
- Falscher Datentyp für Operator oder @Funktion: Zeit/Datum erwartet
- Variant enthält keinen Container

Subject	Valid Till
FindAndReplace - template - do not erase	FEHLER: Falscher Datentyp für Operator oder @Funktion: Zeit/Datum erwartet 22.03.2012 22:43:37
NumberItem	FEHLER: Falscher Datentyp für Operator oder @Funktion: Zeit/Datum erwartet 22.03.2012 22:43:48
Kalaschnikow-Werbung	FEHLER: Falscher Datentyp für Operator oder @Funktion: Zeit/Datum erwartet 26.01.2011 11:39:59
Test1	FEHLER: Falscher Datentyp für Operator oder @Funktion: Zeit/Datum erwartet 22.03.2012 22:43:48

Unser Ziel: Der einfache und sichere Umgang mit Datum-/Zeitwerten in Lotus Notes

Zeiterfassung:

Mitarbeiter:	Bernhard Koehler		
Abteilung:	Development Notes		
Datum:	<input type="text" value="04.10.2007"/> <input type="text" value="16"/>	(Donnerstag in Woche 40)	
Zeitraum:	von: <input type="text" value="18:00"/> <input type="text" value="⌚"/>	bis: <input type="text" value="00:00"/> <input type="text" value="⌚"/>	Dauer: 5,50 h (Pause: 30 Minuten)
Ort:	Boston MA.		
Auftrag:	031 - TimeReg: Berücksichtigung HoursPerMonth		
Auftraggeber:	Telemotive AG		
Tätigkeit:	Programmierung		
Beschreibung:	Anpassung an EST		
Bemerkungen:			

Zeitwerte in Notes: Ganz einfach? Oder extrem kompliziert?

- Einfacher Umgang durch eigenen Feldtyp (Frontend)
- Eigener Datentyp im Backend
- Hilfreiche @Functions verfügbar – die wichtigste heisst @IsTime
- Spezielle Funktionen für den Umgang mit Zeitwerten in LotusScript – die wichtigste heisst IsDate
- Eigener Datentyp in LotusScript
- Eigene Objekte aus der LS-Klassenbibliothek: NotesDateTime und NotesDateRange

- Und eigentlich ...

Zeitwerte in Notes: Ganz einfach? Oder extrem kompliziert?

- Einfacher Umgang durch eigenen Feldtyp (Frontend)
- Eigener Datentyp im Backend
- Hilfreiche @Functions verfügbar – die wichtigste heisst @IsTime
- Spezielle Funktionen für den Umgang mit Zeitwerten in LotusScript – die wichtigste heisst IsDate
- Eigener Datentyp in LotusScript
- Eigene Objekte aus der LS-Klassenbibliothek: NotesDateTime und NotesDateRange

- Und eigentlich ... **ist Datum/Zeit nur eine Zahl!**

Datum/Zeit im Frontend (Feld)

- Eigener Feldtyp
- Bequem zu erstellen, bequem zu bedienen
- Speicherung von Datum oder Zeit oder beides
- Bietet Steuerelemente, Eingabeunterstützung, Vervollständigung

Feld

Name: DateOfBirth

Typ: Datum/Zeit Bearbeitbar

Mehrfachwerte zulassen

Nach Validierung berechnen

Externe Datenquelle

Stil: Notes Kalender-/Uhrzeit-Steuerelement

Steuerelement und Absatz auf gleiche Grundlinie

Größe: Breite 2,499cm Fest (Größe)

Höhe 0,635cm Fest

Tab-Taste: Position in Tab-Reihenfolge

Beim Öffnen Cursor in dieses Feld setzen

Vorgaben von Client

Anzeige Beispiel 06.10.2007

Anzeige Datum

Anzeigen: Monat, Tag, Jahr

Spezial: Jahr 4stellig anzeigen, wenn 21. Jhd.

Kalender: Gregorianisch

Anzeige Uhrzeit

Anzeigen: Stunden, Minuten

Zeitzone:

Bei Eingabe Jahr muss 4stellig eingegeben werden

Monat muss alphabetisch eingegeben werden

Datum/Zeit im Backend (Item)

The screenshot displays a software interface with a table of items and a detailed view of a selected item.

Table of Items:

Name	N...	S...	D...	I...	Value	Modification date	Is Summary	Is
ab \$ConflictAction	1	1	0	18	1	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab \$UpdatedBy	1	1	0	31	CN=Bernhard Koehler/O=BKNotes/C=DE	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab \$WebFlags	1	1	0	17	V	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab Activity	1	1	0	29	Dienstreise	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab AllowedAuthors	2	1	0	19	[Administrator]; [TReditor]	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab AllowedReaders	2	1	0	20	[Administrator]; [TRReader]	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab APPVERSIONNO	1	1	0	13	1.03.05	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
I.2 Break	1	1	0	24	0	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab COSTCENTERDE...	1	1	0	12		15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab COSTCENTERNO	1	1	0	11		15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab Customer	1	1	0	10	Schiedel Gmbh Austria	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab DEPARTMENT	1	1	0	3	Development Notes	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab Description	1	1	0	30	Nussbach - Siegsdorf	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab DOCID	1	1	0	1	E44DFA0FBB6F094DC12573F0005CD431	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab DTSAVED	1	1	0	14	15.02.2008 17:54:24	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
I.2 Duration	1	1	0	7	90	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab EndTime	1	1	0	23	15.02.2008 15:30:00	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab EntryComments	1	1	0	26		15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab FORM	1	1	0	16	TimeRegistration	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab HISTORY	2	1	0	6	15.02.2008 17:53:59: Bernhard Koehler - Erstell...	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab Location	1	1	0	25	Nussbach	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab Order	1	1	0	8	Training on the job 2008-1	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	
ab OrderNo	1	1	0	9	075	15.02.2008 17:54:21	<input checked="" type="checkbox"/>	

Field Value Detail View:

Note Info Readers/Authors Field Value

Type: **ab** Text Name: TRUSER

CN=Bernhard Koehler/O=BKNotes/C=DE

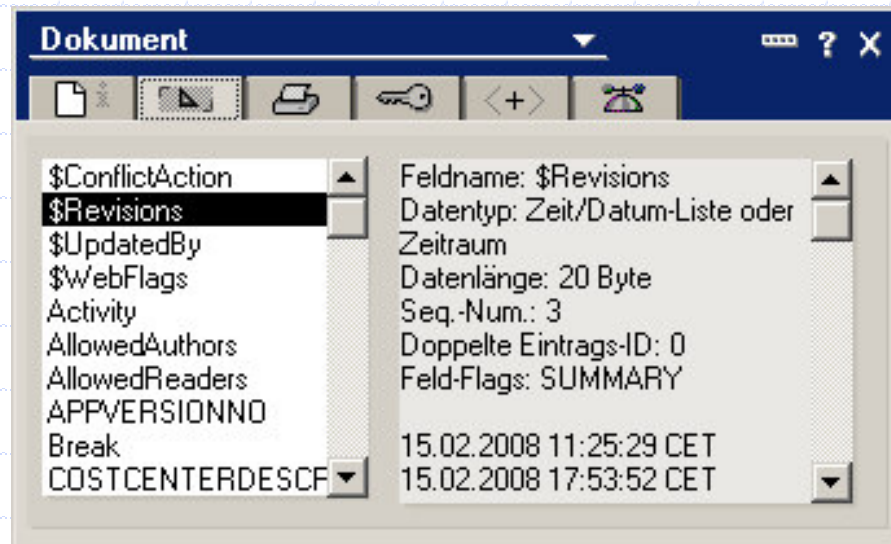
Summary Encrypted Signed Protected Unchanged

15.02.2008 17:54:21 34 bytes 1 0

OK Cancel

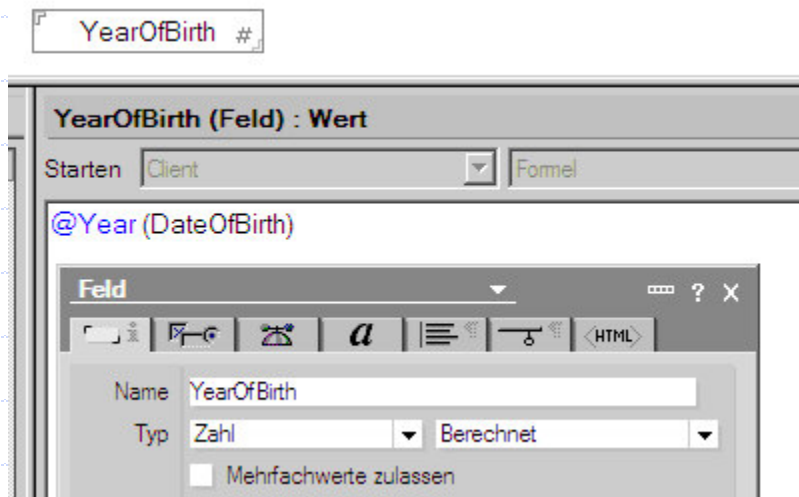
Date/Time-Item im Detail

- Eigener Datentyp
- Beansprucht immer 8 Byte
- Listenfähig
- Date only / Time only: Eine Besonderheit des Frontends
- Speicherung der Zeitzone



Rechnen mit Date/Time

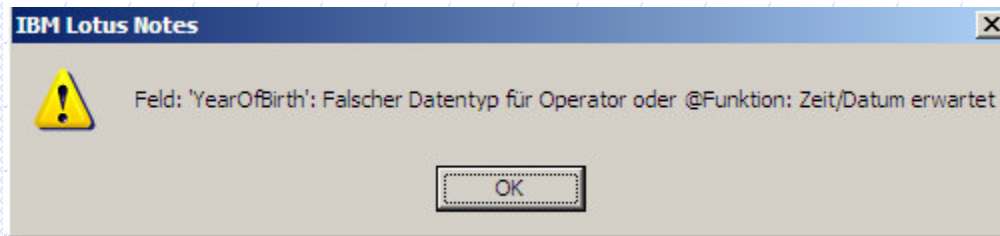
Ermittlung des Geburtsjahres:



Einfach? Zu einfach (gedacht)!

Rechnen mit Date/Time

Wir können kein neues Dokument erstellen – Notes schmettert und sofort eine Fehlermeldung entgegen:



Wie können wir dies verhindern?

Rechnen mit Date/Time

- Der Datentyp Date/Time ist nicht kompatibel mit anderen Datentypen
- Vor Berechnungen immer auf Date/Time prüfen!
- Niemals auf „richtige Werte“ verlassen!
- Niemals mit einem Leerstring vergleichen!

@If (@IsTime (Birthday); @Year (Birthday); YearOfBirth)

- Datums-/Zeitwerte immer im dafür vorgesehenen Datentyp speichern!
Strings sind hierfür tabu!
- Es drohen sonst „internationale Verwicklungen“
Unter „01.12.2008“ versteht ein Domino mit deutschen DT-Settings
etwas ganz anderes als sein britischer Kollege ...
- Und wenn es passiert ist: Nachträgliche Korrekturen lohnen den Aufwand!

Rechnen mit Date/Time

- Umwandlungen (Typwandlung) ist auch mit Formelsprache möglich:
@TextToTime
- Ab R6: @ToTime ... Universeller, da auch DT-Werte akzeptiert werden
- „Gegenrichtung“: @Text
- Leider nicht vorhanden: „@TimeToNumber“

Datumswerte bilden

Datumswert bilden:

@Date (Jahr; Monat; Tag)

@Date ([28.03.2012])

@Date (@Now)

Es wird nur die Datumskomponente zurückgegeben.

@Time arbeitet analog.

Datumswerte anpassen

@Adjust (Ausgangswert; Jahre; Monate; Tage; Stunden; Minuten; Sekunden)

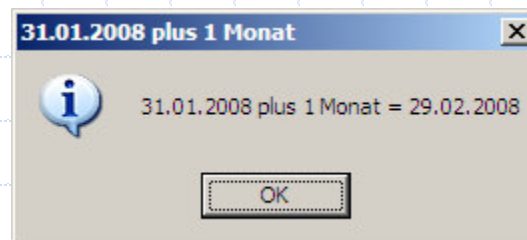
@Adjust (@Today; 0; 0; 1; 0; 0; 0) ergibt das Datum des morgigen Tages

@Weekday (@Adjust (@Today; 0; 0; -100; 0; 0; 0)) ergibt den Wochentag heute vor 100 Tagen.

@Adjust ist „brav“:

StartDate := @Date (2008; 1; 31);

@Prompt ([OK]; "31.01.2008 plus 1 Monat"; "31.01.2008 plus 1 Monat = " + @Text (@Adjust (StartDate; 0; 1; 0; 0; 0; 0)))



Datumswerte anpassen

Weitere Funktionen:

- @Today
- @Yesterday
- @Tomorrow

- Statt @Today: @TextToTime („Heute“)

- @Now mit Zeitkomponente

@Functions arbeiten immer auf der Basis der Zeitzone des Betriebssystems, auf denen sie ausgeführt werden. Im Frontend werden bei Manipulationen Werte aus fremden Zeitzonen an die eigene angepasst!

Beispiele für @Adjust

Letzten Wochentag des aktuellen Monats ermitteln:

```
NextMonth := @Adjust (@Today; 0; 1; 0; 0; 0; 0);
```

```
Ultimo := @Adjust (@Date (@Year (NextMonth); @Month (NextMonth); 1); 0; 0; -1; 0; 0; 0);
```

```
LastWeekday := @Weekday (Ultimo);
```

```
@Prompt ([OK]; "Letzter Wochentag"; @Word ("Sonntag Montag Dienstag Mittwoch Donnerstag Freitag  
Sonnabend"; " "; LastWeekday))
```

Ermittlung Quartalsdaten:

```
@If (!@IsTime (Datum); @Return (""); "");
```

```
month1 := (@Integer ((@Month (Datum) - 1) / 3) + 1) * 3;
```

```
day1 := @Date (@Year (Datum); month1 - 2; 1);
```

```
temp := @Adjust (day1; 0; 3; 0; 0; 0; 0);
```

```
REM {Und jetzt wieder einen Tag zurück ...};
```

```
day2 := @Adjust (temp; 0; 0; -1; 0; 0; 0);
```

```
@Prompt ([Ok]; "Quartalsdaten"; "Erster Tag = " + @Text (day1) + "    Letzter Tag = " + @Text (day2))
```

Differenzberechnungen

Altersberechnung (Problem: Geburtstag > heute?)

```
_Age1 :=@If (@IsTime (DateOfBirth); (@Year (@Today) - @Year (DateOfBirth)); @Return (""));
```

```
_Age2 := @If (@Adjust (DateOfBirth; _Age1; 0; 0; 0; 0; 0) > @Today; _Age1 - 1; _Age1);
```

```
_Age2
```

Differenz in Tagen:

```
@If (@IsTime (DateOfBirth); (@Today - DateOfBirth) / 86400; "")
```

Differenz in Stunden:

```
@If (!@IsTime (StartTime) | !@IsTime (EndTime); @Return (""); "");
```

```
(EndTime - StartTime) / 3600
```

Berechnung der Kalenderwoche

@If (!@IsTime (Datum); @Return (""); "");

Wochenstart := @Adjust(Datum; 0; 0; -@Weekday (Datum) + @If
(@Weekday (Datum) = 1; -5; 2); 0; 0; 0);

ErsterJanuar := @Date (@Year (Wochenstart); 1; 1);

ErsterWochentag := @Weekday (ErsterJanuar);

TagImJahr := (Wochenstart - ErsterJanuar) / (60 * 60 * 24);

Kalenderwoche := 1 + @Round (TagImJahr / 7);

KW := @If ((ErsterWochentag *= 2 : 3 : 4) & (Kalenderwoche = 53); 1;
Kalenderwoche);

KW

Erzeugung von Datumslisten

Wichtig beispielsweise für

- Kalenderdarstellung wiederkehrender Termine
- Verfügbarkeitprüfungen

Sieht „tricky“ aus, ist aber nur die konsequente Nutzung

- der Konvertierungsfunktionen
- der mächtigen @Functions zur Listenbearbeitung

```
@If (!@IsTime (StartDate) | !@IsTime (EndDate); @Return (""); "");
```

```
@Explode (@TextToTime (@Text (StartDate) + "-" + @Text (EndDate)))
```

Datumswerte in Ansichten

Datumswerte können problemlos in Ansichten dargestellt werden (auch berechnet!), wenn man bestimmte Regeln beachtet:

- Validierung der Werte
- Darstellung in der gewünschten Form (z.B. mit @Date, da wir unser Datenmaterial nie im Voraus kennen können).

Subject	Geburtstag + 1 Jahr
Igor	29.01.85
Kuno	FEHLER: Falscher Datentyp für Operator oder @Funktion: Zeit/Datum erwartet
Stefan	11.07.91 22:05:00

Datumswerte in Ansichten

- Möglichst vermeiden: Der Bezug auf den aktuellen Zeitpunkt
 - @Today
 - @Now
 - @Yesterday, @Tomorrow

Dies erzwingt den Server permanent zur Aktualisierung des Index!
Die Datenmenge ist hier entscheidend, ein generelles Verbot ist nicht angebracht.

-> User können irritiert werden, weil bei derartigen Ansichten immer der Aktualisierungskringel angezeigt wird!

Datumswerte in Ansichten

- Möglichst vermeiden: Der Bezug auf den aktuellen Zeitpunkt

- @Today
- @Now
- @Yesterday, @Tomorrow

Dies erzwingt den Server permanent zur Aktualisierung des Index!
Die Datenmenge ist hier entscheidend, ein generelles Verbot ist nicht angebracht.

-> User können irritiert werden, weil bei derartigen Ansichten immer der Aktualisierungskringel angezeigt wird!

- Dies ermöglicht jedoch einen „dirty trick“:

- versteckte Spalte mit einer der o.g. Datumsfunktionen
- jedes Öffnen der DB aktualisiert die Ansicht neu
- damit funktioniert plötzlich auch @UserName in Ansichten.

ACHTUNG: Übel! Dieses Verfahren ist unsicher und killt Performance!



**Wir verlassen die @Functions:
LotusScript und neue Möglichkeiten!**

Das „magische“ Datum:

Wer kennt die Bedeutung des

30. Dezember 1899

In Bezug auf Lotus Notes / Domino?

Zusatzfrage: Was geschah am 10. Oktober 1582 in Rom?

Das Notes Date/Time-Item ist „nur“ eine Zahl!

Fangen wir mit der „einfachen“ Frage an:

Am 10. Oktober 1582 fand in Rom garantiert gar nichts statt – durch die Umstellung vom Julianischen auf den Gregorianischen Kalender folgte auf den 4. Oktober gleich der 15. Oktober.

Und der 30. Dezember 1899?

Das Notes Date/Time-Item ist „nur“ eine Zahl!

Wir bauen uns einen Button in einer simplen Maske:

```
Sub Click (Source As Button)
```

```
    Dim ws As New NotesUIWorkspace
```

```
    Dim uidocCurrent As NotesUIDocument
```

```
        Set uidocCurrent = ws.CurrentDocument
```

```
        uidocCurrent.Document.Day0 = Cdat (0)
```

```
        Call ws.ViewRefresh
```

```
End Sub
```

Das Notes Date/Time-Item ist „nur“ eine Zahl!

Das Resultat:

Tag 0 setzen

Der Tag 0: 30.12.1899

Wir sehen:

- Datums-/Zeitwerte werden als Datumsseriennummer gespeichert
- „Tag 0“ ist der 31.12.1899
- Der ganze Teil der Datumsnummer repräsentiert den Tag
- Der gebrochene Teil repräsentiert die Uhrzeit.
- 39504,75 entspricht dem 26. Februar, 18:00 Uhr.

Zugriff auf Date/Time-Items mit LotusScript

- Die LotusScript Klassenbibliothek bietet ein eigenes Objekt: NotesDateTime
- Wir wissen aber: Date/Time-Items beinhalten „eigentlich“ nur (rationale) Zahlen.
- Date/Time-Items können in Variants vom Typ 7 (Date/Time) eingelesen werden bzw. durch solche belegt werden
- Vorab: Das Date/Time-Handling via Variants vereinfacht die Programmierung sehr stark, es bietet aber keinen Zugriff auf die Manipulation von Zeitzonen.

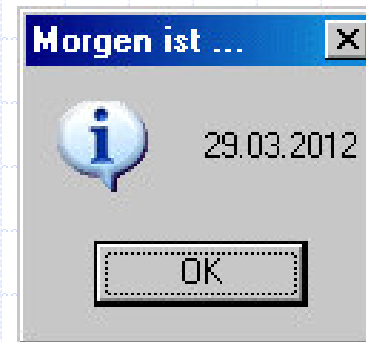
Einfache Beispiele:

```
Dim vDate As Variant
```

```
vDate = Today
```

```
vDate = vDate + 1
```

```
MessageBox Cstr (vDate), MB_ICONINFORMATION, "Morgen ist ..."
```



oder

```
NotesDocument.Datum = Today
```

```
NotesDocument.Datum = NotesDocument.Datum (0) + 1
```

Wichtig: Validierung!

Wie in der Formelsprache, ist Validierung auch mit LS unabdingbar!

Aus der DesignerHelp:

IsDate (*expr*)

Elements

expr

Any expression.

Return value

IsDate returns TRUE (-1) if *expr* is any of the following:

- A Variant value of DataType 7 (Date/Time)
- A Variant value of type String, where the string represents a valid date/time value
- A String value representing a valid date/time value

Otherwise IsDate returns FALSE (0).

Validierung: Houston, wir haben ein Problem!

IsDate akzeptiert auch Strings, die wie ein Datum aussehen (mit CDate konvertiert werden können) ...

Isdate (Today) = True

Isdate („26.02.2008“) = True

Trifft unser Code dann doch auf einen String ...

Was tun?

Validierung mit LotusScript

Wir verwenden nach Isdate die Datatype function!

Rückgabewerte:

7 Date/Time

12 Variant list or array

2,048 List

8,192 Fixed array

8,704 Dynamic array

- Ist der Wert Skalar? -> Datatype muss 7 sein
- Ist der Datatype > 8192 (fixed or dynamic array) -> Element 0 muss Datatype 7 sein.
- Für Listen: Eigene Behandlung (analog)

Validierung mit LotusScript

```
Function ValidateDTValue (vGivenValue As Variant) As Integer
    On Error Goto ErrorRoutine
    'Default return value:
    ValidateDTValue = False
    If IsScalar (vGivenValue) Then
        If Datatype (vGivenValue) = 7 Then
            ValidateDTValue = True
        Else
            ValidateDTValue = False
        End If
    End If
    Exit Function
    If Datatype (vGivenValue) > 8192 Then
        If Datatype (vGivenValue (0)) = 7 Then
            ValidateDTValue = True
        Else
            ValidateDTValue = False
        End If
    End If
    Exit Function
End If
Exit Function
ErrorRoutine:
    ValidateDTValue = False
    Call ErrorHandler ("ValidateDTValue")
    Exit Function
End Function
```

Validierung mit LotusScript: **Noch eine Falle!**

Validierung im Backend – mit ungültiger Eingabe im Frontend:

Subject: 'Test validation'

Date: 'huhu'

Remarks: 'Bad gambling with DT fields ...'

```
Set doc = ws.CurrentDocument.Document
```

```
If Isdate (doc.Date (0)) = False Then  
  MsgBox "Wrong input!"  
Exit Sub
```

```
Else  
  MsgBox "Input OK"  
End If
```



Validierung mit LotusScript: Noch eine Falle!

WARUM?

The screenshot shows a LotusScript debugger window with the following content:

```
If Isdate (doc.Date (0)) = False Then  
    MessageBox "Wrong input!"
```

The debugger interface includes tabs for Breakpoints, Variablen, Ausgabe, and Aufrufe. The 'Variablen' tab is active, displaying a tree view of the current scope:

- ITEMS [NOTESITEM()]
 - [0] ["Subject", 1280, 17, False, False, True,...]
 - [1] ["Date", 256, 4, False, False, True, False,...]
 - NAME "Date"
 - TYPE 256
 - VALUES
 - VALUELENGTH 4
 - ISENCRYPTED False
 - ISSIGNED False
 - ISSUMMARY True
 - ISPROTECTED False
 - PARENT [False, 0:00:00, 26.03.2012 17:25:05,...]
 - TEXT "ERROR: "

Validierung mit LotusScript: **Noch eine Falle!**

Lösung:

Ein Item mit gespeichertem Fehlerstatus ist bei Frontend-Abfrage kein gültiges Item.

```
If Isarray (doc.Date) = False Then
    MessageBox "Totally wrong input!"
    Exit Sub
Else
    If Isdate (doc.Date (0)) = False Then
        MessageBox "Wrong input!"
        Exit Sub
    Else
        MessageBox "Input OK"
    End If
End If
```

Erstellung, Konvertierung

Ähnlich wie in Formelsprache:

- Cdat / @ToTime / @TextToTime
- Datenummer (@Date)
- TimeNumber (@Time)
Hinweis:
 $vDate = \text{DateNumber}(2008, 2, 26) + \text{TimeNumber}(16, 0, 0)$
ergibt
26.02.2008 16:00:00 Uhr
- Year, Month, Day, Hour, Minute, Second
- Today, Now
- Tomorrow, Yesterday fehlen
(Egal! Tomorrow = Today + 1!)

- Und @Adjust? Später mehr dazu ...

Schleifen über das Datumswerte

- Die Datumsseriennummer kommt uns entgegen!
- Wir können einfach über Datumswerte iterieren
- Wir verwenden hierfür wieder Variants vom Typ 7
- Wir können das Ergebnis einfach einem NotesItem zuweisen
- Beispiel:

Schleifen über das Datumswerte

Sub Click (Source As Button)

```
Dim ws As New NotesUIWorkspace  
Dim docCurrent As NotesDocument  
Dim aDateLoop (0) As Variant  
Dim vLoop As Variant  
Dim vDateList As Variant
```

```
On Error Goto ErrorRoutine  
Set docCurrent = ws.CurrentDocument.Document  
If Not (Isdate (docCurrent.StartDate (0))) Or Not (Isdate (docCurrent.EndDate (0))) Then  
    MsgBox "Datumswert fehlt!"  
    Exit Sub
```

```
End If  
vDateList = docCurrent.StartDate  
For vLoop = docCurrent.StartDate (0) + 1 To docCurrent.EndDate (0) Step 1  
    aDateLoop (0) = vLoop  
    vDateList = Arrayappend (vDateList, aDateLoop)  
Next  
docCurrent.DateList = vDateList  
Call ws.ViewRefresh  
Exit Sub
```

ErrorRoutine:

```
Call ErrorHandler ("LS2 - Datumsliste füllen")  
Exit Sub
```

End Sub

Summierung, Aufrechnung

Wenn Datums-/Zeitwerte vieler Dokumente aufsummiert werden sollen oder als Bruchteile von Stunden oder Tagen aufgeführt werden müssen:

- Egal, ob Formelsprache oder LotusScript:
- Immer mit der kleinsten möglichen (sinnvollen) Einheit rechnen
- Normalerweise: Zeitspannen werden in Minuten gespeichert und erst später in Bruchteile von Stunden oder Tagen umgerechnet
- Rundungsfehler vermeiden!

von - bis | Beschreibung | Stunder | Kunde | Auftrag | Tätigkeit

0.00

Objekte Referenz

- Zeiterfassung\Nach KW (Globals)
- Zeiterfassung\Nach KW (Ansicht)
 - ◆ Ansichtsauswahl
 - ◆ Maskenfomel
 - ▲ ...

Stunden (Spalte) : Spaltenwert

Anzeige Einfache Funktion Feld Formel

@If (Status = "Discarded"; 0; (Duration / 60))

Und @Adjust??

Kommt gleich!

Vorab:

Es scheint ganz einfach:

Der erste Tag des Monats in 13 Monaten:

Dim vDate As Variant

vDate = Today

vDate = Datenumber (Year (vDate), Month (vDate) + 13, 1)

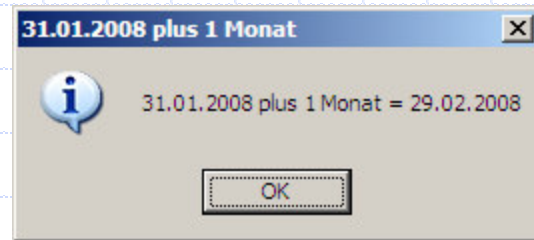
Ergebnis: Wie erwartet der 01.03.2009

Die NotesDateTime-Klasse

Wir erinnern uns an unser Formelsprachen-Beispiel:

```
@Prompt ([OK]; "31.01.2008 plus 1 Monat"; "31.01.2008 plus 1 Monat = " +
```

```
@Text (@Adjust (StartDate; 0; 1; 0; 0; 0; 0)))
```



Da müssten wir mit LotusScript „pur“ viel Handarbeit anlegen – oder wir nutzen jetzt doch die NotesDateTime-Klasse ...

Die NotesDateTime-Klasse

Die NotesDateTime-Klasse bietet zahlreiche Eigenschaften und Methoden:

Properties

[DateOnly](#)
[GMTTime](#)
[IsDST](#)
[IsValidDate](#)
[LocalTime](#)
[LSGMTTime](#)
[LSLocalTime](#)
[Parent](#)
[TimeOnly](#)
[TimeZone](#)
[ZoneTime](#)

Methods

[AdjustDay](#)
[AdjustHour](#)
[AdjustMinute](#)
[AdjustMonth](#)
[AdjustSecond](#)
[AdjustYear](#)
[ConvertToZone](#)
[New](#)
[SetAnyDate](#)
[SetAnyTime](#)
[SetNow](#)
[TimeDifference](#)
[TimeDifferenceDouble](#)

Die NotesDateTime-Klasse

- Viele Zuweisungen und Rückgaben erfolgen via String – dies sollte (wenn immer möglich) vermieden werden. Die Gründe wurden bereits genannt
- Zuweisung ideal nur mit gleichem Datentyp:
Set NotesDateTime = NotesItem.DateTimeValue
oder
Set NotesItem.DateTimeValue = NotesDateTime
- Wenn immer möglich: Finger weg von Stringkonstrukten wie
Set NotesDateTime = New NotesDateTime („26.02.2008“)
(das versteht kein amerikanischer Server ...)

Die NotesDateTime-Klasse: Unser @Adjust

Sub Click (Source As Button)

```
Dim ws As New NotesUIWorkspace
Dim docCurrent As NotesDocument
Dim itemDT As NotesItem
Dim dtCalculation As NotesDateTime

On Error Goto ErrorRoutine

Set docCurrent = ws.CurrentDocument.Document

If Not (Isdate (docCurrent.StartDate (0))) Then
    MessageBox "Datumswert fehlt!"
    Exit Sub
End If

docCurrent.EndDate = docCurrent.StartDate

Set itemDT = docCurrent.GetFirstItem ("EndDate")
If itemDT Is Nothing Then
    MessageBox "Datumswert fehlt!"
    Exit Sub
End If

Set dtCalculation = itemDT.DateTimeValue
Call dtCalculation.AdjustMonth (1,True)
Set itemDT.DateTimeValue = dtCalculation

Call ws.ViewRefresh

Exit Sub

ErrorRoutine:
Call ErrorHandler ("LS2 - Datumsliste füllen")
Exit Sub

End Sub
```

Startdatum:

31.01.2012 16

Startdatum plus 1 Monat

Plus ein Monat:

29.02.2012 16

Die NotesDateTime-Klasse: Übler Fehler

Mit herzlichen Dank an Ulrich „eknorl“ Krause, der mich auf diesen Bug hingewiesen hat:

```
Sub Click (Source As Button)
```

```
    Dim dt As NotesDateTime
```

```
    Dim y As Integer
```

```
    On Error Goto ErrorRoutine
```

```
    y = 1700
```

```
    Set dt = New NotesDateTime (Datenumber (y, 3, 1))
```

```
    Call dt.AdjustDay (-1)
```

```
    If Day (dt.DateOnly) = 29 Then
```

```
        Print "Schaltjahr: " & Cstr (y)
```

```
    End If
```

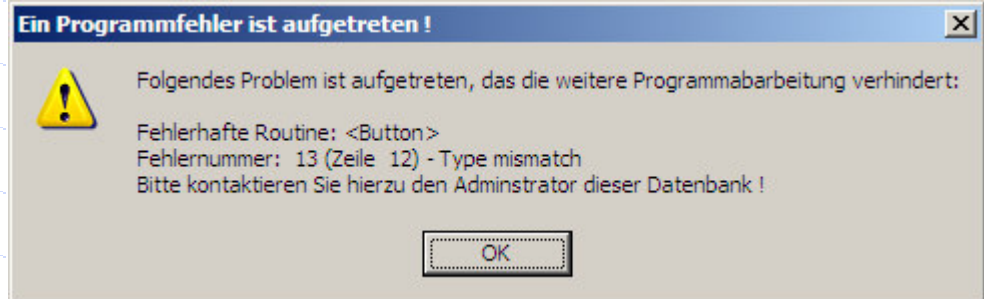
```
    Exit Sub
```

```
ErrorRoutine:
```

```
    Call ErrorHandler ("Button")
```

```
    Exit Sub
```

```
End Sub
```



Die NotesDateTime-Klasse: Übler Fehler

Eine leichte Anpassung macht das nicht besser ...

```
Sub Click (Source As Button)

    Dim dt As NotesDateTime
    Dim y As Integer

    On Error Goto ErrorRoutine

    y = 1700

    Set dt = New NotesDateTime (Datenumber (y, 3, 1))
    Call dt.AdjustDay (-1)
    Print Left$ (dt.DateOnly, 2)

    If Left$ (dt.DateOnly, 2) = "29" Then
        Print "Schaltjahr: " & Cstr (y)
    End If

    Exit Sub

ErrorRoutine:
    Call ErrorHandler ("Button")
    Exit Sub

End Sub
```



Mit LotusScript pur klappt es dann doch ...

```
Sub Click (Source As Button)
```

```
    Dim iYear As Integer  
    Dim vDate As Variant
```

```
    On Error Goto ErrorRoutine
```

```
    iYear = 1700
```

```
    vDate = Datenummer (iYear, 3, 1)  
    vDate = vDate - 1
```

```
    If Day (vDate) = 29 Then
```

```
        MsgBox "Schaltjahr: " & Cstr (iYear), "Schaltjahr-Status"
```

```
    Else
```

```
        MsgBox "KEIN Schaltjahr: " & Cstr (iYear), "Schaltjahr-Status"
```

```
    End If
```

```
    Exit Sub
```

```
ErrorRoutine:
```

```
    Call ErrorHandler ("Button")
```

```
    Exit Sub
```

```
End Sub
```



Zeitzone berücksichtigen: Nur mit NotesDateTime!

- Werden Date/Time-Daten über das Frontend gespeichert: Es gelten immer die Vorgaben des Betriebssystems
- Werden Date/Time-Daten über das Backend gespeichert und Date/Time-Variants benutzt: Es gelten auch hier immer die Daten des Betriebssystems, auf dem der Client oder Server läuft.
- Wie können Date/Time-Werte in einer anderen Zeitzone als der (Client-/Server-)eigenen gespeichert werden?

Zeitzone berücksichtigen: Nur mit NotesDateTime!

Jetzt brauchen wir die NotesDateTime-Klasse wirklich!

Changes the [TimeZone](#) and [IsDST](#) properties as specified.

Defined in

[NotesDateTime](#)

Syntax

Call *notesDateTime*.**ConvertToZone**(*newzone*, *dst*)

Parameters

newzone

Integer. A time zone. Could be negative.

dst

Boolean. Indicates whether daylight-saving time is in effect (True) or not (False).

Usage

These changes affect the [ZoneTime](#) property.

The [GMTTime](#) and the [LocalTime](#) properties do not change.

This method has no effect on an invalid [TimeZone](#) (because the DateTime object lacks the date or the time). The invalid TimeZone continues to return 0.

Zeitzonen berücksichtigen: Nur mit NotesDateTime!

Wir berechnen die Gelsenkirchener Zeit auf die Moskauer Zeit um:

```
Sub Click (Source As Button)
```

```
    Dim ws As New NotesUIWorkspace  
    Dim docCurrent As NotesDocument  
    Dim dtObject As NotesDateTime  
    Dim itemDT As NotesItem  
    Dim vDT As Variant
```

```
    On Error Goto ErrorRoutine
```

```
    Set docCurrent = ws.CurrentDocument.Document  
    Set itemDT = docCurrent.GetFirstItem ("DTValue")
```

```
    If itemDT Is Nothing Then  
        MsgBox "Item 'DTValue' not found!"  
        Exit Sub
```

```
    End If
```

```
    Set dtObject = itemDT.DateTimeValue  
    Call dtObject.ConvertToZone (dtObject.TimeZone + 2, True)  
    Set itemDT.DateTimeValue = dtObject
```

```
    Call docCurrent.Save (True, False, True)
```

```
Exit Sub
```

```
ErrorRoutine:
```

```
    Call ErrorHandler ("Click")
```

```
    Exit Sub
```

```
End Sub
```

Fragen? Immer her damit!

- Für Fragen stehe ich jederzeit gerne zur Verfügung:
 - Jetzt
 - Abseits der Session
 - per eMail
 - und natürlich in Foren wie AtNotes oder DominoForum
- Es sind noch viele Punkte offen, die in so kurzer Zeit nicht abzuklären sind. Zum Beispiel:
Wozu taugt @BusinessDays – und warum ist das auch nur ein Notbehelf? Und wie berechnet man wirklich die Feiertage anwenderbezogen?