smartiX consulting gmbh

LotusScript ohne Grenzen

Your partner for *smart* solutions

Veit Florian Lier (CEO)
Feb, 22th 2006

smartiX consulting gmbh

VWR INTERNATIONAL | Deutsche Bank | DaimlerChrysler | Henkel | LBBW | DC | ALTANA | smart | Amersham Biosciences | Volksbanken | era | IBM

IBM Business Partner | .domino Lotus | DNUG User Group | B-KH

Gründungsmitglied des deutschen Notes Expertenforums

Projects	Products	Services	Experts

Groupware und Lotus Notes

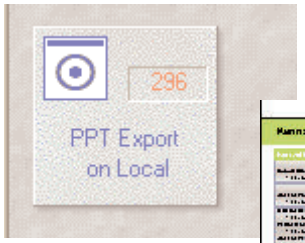
Consulting and IT services provider; high quality, technology-driven business solutions; Stuttgart (HQ); founded 1999; 5+ employees; strategic alliances; over 100 projects

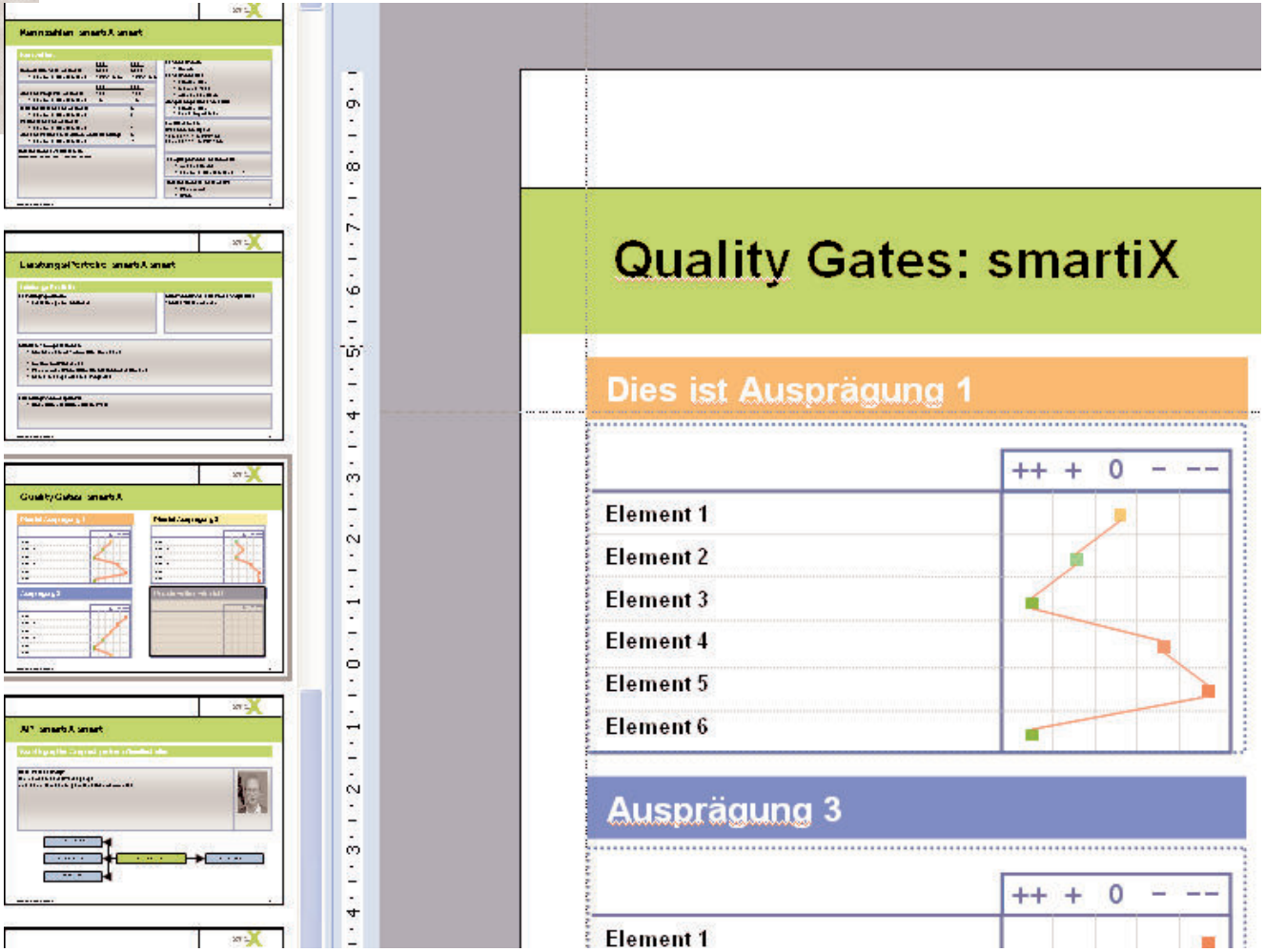
- 31 Jahre – dipl.-oec. (Universität Hohenheim)
- Gründer der smartiX consulting gmbh (1999)
- Notes-Entwicklung / Administr. Seit 1995
- Ca. 40 Zertifizierungen (CLS, CLP, PCLP)
- Domino.Doc, LotusWorkflow....
- Mitglied bei Penumbra, B-KH, ...
- Seit 2005 Workplace Designer ;-)

- Nice Colours...
  - Management liebt Farben und Statistiken
  - Diese müssen von Angestellten zugeschickt werden...

- Lotus Notes...
  - Ausgezeichnete Plattform für die Erfassung und Verwaltung von Daten aber...
  - Bescheidene Darstellungs-Möglichkeiten... Insbesondere:
  - Keine Schaubilder

- MS PowerPoint
  - DeFacto Industrie-Standard für Bilder und graphische Darstellung von „Trends" und Übersichten...
  - Reine Präsentations-Ebene...
  - ➔ Wie kommen die Daten aus Notes in PPT ???


296
PPT Export
on Local


Quality Gates: smartiX
Dies ist Ausprägung 1
++ + 0 - --
Element 1
Element 2
Element 3
Element 4
Element 5
Element 6
Ausprägung 3
++ + 0 - --
Element 1

- Export von Text
  - Allgemeine Informationen (Datum – Autor - ....)
  - Datenbank-Inhalte wie z. B. Felder (Firmenname)

- Export als Graphik
  - Erstellung von Organigrammen oder
  - Text-Boxen, Kreisen, PPT-Standard-Elementen

- Statistiken, Schaubildern
  - Typischerweise erstellt als MS Excel und der Import der Ergebnisse als „Gestaltungs-Element“

- Erstellung COM-Object (PPT)
- Zuweisung von File...
  - Ersetzen von Platzhaltern
  - Erstellung von neuen Objekten
  - ...
- Export von Informationen ....
  - Erstellung von Excel-Dateien
  - Speichern und Kopieren....
- Importieren von „externen Quellen“
- Schliessen aller offenen Objekte

smartiX consulting gmbh

```
'### Initialize PowerPoint...
    Dim PPT              As      Variant          '### PowerPoint-Objekt
    Dim MyPresentation   As      Variant          '### The Presentation itself
    Dim MySlideNumber    As      Integer          '### Current Slide-Number
'### Creation of an PPT OLEObject
    Set PPT              = CreateObject("PowerPoint.Application")
    PPT.Visible          = True
'### Creation of a Presentation
    str_TemplateName     = DetachTemplate("PartnerProfile")
    Set MyPresentation   = PPT.Presentations.Open(str_TemplateName, 0, False) 'entspricht der Konstanten:
```

Export von Texten

```
'### Replace all values in all slides...
  Forall sld In MyPresentation.Slides                '### Each Slide
    Forall Sha In sld.Shapes                         '### Each Squares, Text-Boxes.....
'### Print History
      str_Further          = str_Further & "||"
      Print str_Further & " processing: " &  sld.Name & " - " & Sha.Name
      If Sha.TextFrame.HasText Then                  '### Only the text.....
'### Replace Text
        For i = 0 To Ubound(va_To)
          res              = Evaluate(va_From(i), doc_C)
          str_res          = ReplaceSubstring(res(0), "~#+", Chr(13) & " ")
          str_res          = Trim(ReplaceSubstring(str_res, "...", Chr(13) & " "))
          Sha.TextFrame.TextRange.Replace va_To(i), str_res, False      '### Replace String by String
        Next
      End If
    End Forall
  End Forall
```


smartiX consulting gmbh

Erstellen einer Box und Formatierung

```
        While Not doc_Child Is Nothing
'### Get the last slide
        MySlideNumber           = MyPresentation.Slides.Count
'### Copy it
        PPT.ActiveWindow.View.GotoSlide MySlideNumber
        PPT.ActiveWindow.Selection.SlideRange.Duplicate
'### Run back
        With PPT.ActiveWindow.View
            .GotoSlide MySlideNumber
        End With
'### Insert MySelf Frame——
        PPT.ActiveWindow.Selection.SlideRange.Shapes.AddShape(1, 290, 410, 180, 25).Select
'### Name it myself
        With PPT.ActiveWindow.Selection.ShapeRange
            .Fill.Visible           = True
            .Fill.Solid
'### smartiX
            .Fill.ForeColor.RGB     = RGB(142, 190, 000)
            .Name                   = "MySelf"
        End With
'### Set the propertiies..
        With PPT.ActiveWindow.Selection.TextRange
            .Text                   = getFullName_A(doc_Child)
            With .Font
                .Name               = "Verdana"
                .Size               = 10
                .Bold               = 1
                .Italic             = 0
                .Underline          = 0
                .Shadow             = False
                .Emboss             = False
                .BaselineOffset     = 0
            End With
        End With
```


B-KH

smartiX consulting gmbh

Erstellen einer Box und Formatierung

```
'### Get the Position related to the Mean-RectAngle...
    i_Left              = 100
    i_top               = 410
    i_Width             = 150
    i_Height            = 25

'### One above , one the same height and one below...
    If i_Count < 4 Then
        i_top           = i_top - ((i_Count-2)*45)
    Else
        i_Left          = 520
        i_top           = i_top - ((i_Count-4)*45)
    End If

'### Draw the rectangle....
    PPT.ActiveWindow.Selection.SlideRange.Shapes.AddShape(1, i_Left, i_top, i_Width, i_Height).Select

'### Set the color and the name to identify later on....
    With PPT.ActiveWindow.Selection.ShapeRange
        .Fill.Visible      = True
        .Fill.Solid
        .Name          = "Star" & Cstr(i_Count)
        .Fill.ForeColor.RGB = RGB(160, 194, 220)
    End With
```

Dynamische Platzierung der Boxen in der PPT-Folie über iterative Berechnungen
Vergabe von eindeutigen Namen („Star1“, „Star2“) um zu einem späteren Zeitpunkt zu referenzieren...
RGB-Werte über folgende Funktion

```
Function RGB(r As Long, g As Long, b As Long) As Long
    RGB = r + 256*g + 65536*b
End Function
```

```
'### Create a new arrow... (EllBow = 2, Positions variable...
    PPT.ActiveWindow.Selection.SlideRange.Shapes.AddConnector(2, 258, 366, 32, 56.5).Select
    With PPT.ActiveWindow.Selection.ShapeRange
'### Make a nice arrow..
        .Line.EndArrowheadStyle = 2
        .Flip 0
        .Flip 1
    End With

'### The first three draw from left to right side
    If i_Count < 4 Then
        PPT.ActiveWindow.Selection.ShapeRange.ConnectorFormat.BeginConnect PPT.ActiveWindow.Selection.SlideRange.Shapes("MySelf"), 2
        PPT.ActiveWindow.Selection.ShapeRange.ConnectorFormat.EndConnect PPT.ActiveWindow.Selection.SlideRange.Shapes("Star" & Cstr(i_Count)), 4
    Else
'### The next three vice versa
        PPT.ActiveWindow.Selection.ShapeRange.ConnectorFormat.BeginConnect PPT.ActiveWindow.Selection.SlideRange.Shapes("MySelf"), 4
        PPT.ActiveWindow.Selection.ShapeRange.ConnectorFormat.EndConnect PPT.ActiveWindow.Selection.SlideRange.Shapes("Star" & Cstr(i_Count)), 2
    End If
'### Set the line-strength to 2,5
    PPT.ActiveWindow.Selection.ShapeRange.Line.Weight = 2.5

    Exit Function
ErrorHandler:
    Call Error_Set(Cstr(Erl()), Cstr(Lsi_info(2)), Cstr(Lsi_info(3)), Cstr(Lsi_info(12)), Error, Err())
    Resume Next
End Function
```

Hier können wir über Namen (nicht dokumentiert in MS) wieder die einzelnen Elemente verbinden und dynamisch auf diese zugreifen...

Zuerst schreiben wir die Daten in ein definiertes Excel_SpreadSheet

```
'### Wechseln zum richtigen Slide von Excel und Aufruf unserer Ansicht
    Print "collecting data for ""Searches pro Ebene"""
    Set xlsheet                     = xlwb.WorkSheets("Slide3")
    Set view_Exp                    = db_Curr.GetView("va_projname")
    Set nav                         = view_Exp.CreateViewNav
    Set ve                          = nav.GetFirst
'### ViewEntries zum durchlaufen der Excel-Inhalte da schneller !!!
    While Not ve Is Nothing
        If ve.IsCategory Then
            If ve.IndentLevel = 0 Then
                xlsheet.Cells(i_Res, 1).value       = ve.ColumnValues(1)
                xlsheet.Cells(i_Res, 2).value       = ve.ColumnValues(0)
                i_Res = i_Res + 1
            End If
        End If
        Set ve                  = nav.GetNext(ve)
    Wend
```

Und dann folgt die optimale Art zu importieren... Copy & Paste
Andere Methoden führen zu Problemen mit Ausrichtung und Schriftgrößen

```
'### Import Picture Slide 21
    Set xlsheet                 = xlwb.WorkSheets("Slide10")
    xlsheet.ChartObjects("Chart 5").Copy
    PPT.ActiveWindow.View.GotoSlide 21
    PPT.ActiveWindow.View.Paste
'### Import Picture Slide 22
    Set xlsheet                 = xlwb.WorkSheets("Slide11")
    xlsheet.ChartObjects("Chart 3").Copy
    PPT.ActiveWindow.View.GotoSlide 22
    PPT.ActiveWindow.View.Paste
```

- Best-Practice:
  - Debuggen in Script-Editor in PPT
  - Msgbox für die Identification der Parameter (anderer Aufbau)
  - Web-Recherche für evtl. missing parts
  - Kopieren des erstellten Codes... %REM... %ENDREM
  - Und Zeile für Zeile entkommentieren

```
ActiveWindow.Selection.SlideRange.Shapes.AddShape(msoShap
eRectangle, 24#, 252#, 42#, 72#).Select
    With ActiveWindow.Selection.ShapeRange
        .Fill.Visible = msoTrue
        .Fill.Solid
        .Fill.ForeColor.RGB = RGB(255, 204, 0)
        .Fill.Transparency = 0#
    End With
```

Herausforderung: msoTrue, RGB...

```
_LUP:=@DBLookup(„";"NoCache";_DBNAME;_VIEW;_KEY;_RESULT);
@If(@IsError(_LUP);@Text(_LUP);_LUP)
```

- Old fashioned... Aber hilfreich da Anwendungen immer funktionieren!!

```
_A:=@Command([FileSave]);
_B:=@Prompt([OK];_TITLE;_TEXT);
_C:=@Command([FileOpenDatabase];_DBNAME);
_D:=@Command.....
```

- Nützliche Variante um auf diese Weise die Reihenfolge zu manipulieren

- Immer 2 Fragen: Wie fange ich ab – und wie dokumentiere ich wo es passiert??
- Wohin protokolliere ich am Besten?
- Fehler im Fehlerhandling sind „ungeschickt“ ;-)
- Typisch / Konventionell

```
On Error Goto ErrorHandler


Exit Function
ErrorHandler:
Msgbox(„Error in agent MyName in Function TEST in Line „ & Erl)
Resume Next
End Sub
```

- Besser: undokumentierte Feature
- Call Error_Set(Cstr(Erl()), Cstr(Lsi_info(2)), Cstr(Lsi_info(3)), Cstr(Lsi_info(12)), Error, Err())

```
Function Error_Set(str_Line As String, str_Function As String, str_Module As String, str_CalledBy As String, str_Error As
      String, str_ErrCode As String) As Integer
'#####################################
'#  (c) Florian Lier, 24.05.2005
'#####################################
'# @Input (str_Line as string)              The line of the code the error happened
'# @Input (str_Function as string)          The name of the function raising the error
'# @Input (str_Module as string)            The name of the module raising the error
'# @Input (str_CalledBy as string)          The name of the element calling the function / sub / property / class
'# @Input (str_Error as string)             The description of the error
'# @Input (str_ErrorCode as string)         The LS_code of the error
'# @Return (Error_Set as Integer)           Proofs if the function has been completed successfully (True) or not (False)
'# @Description This function uses undocumented calls to track error-logging called LSI_INFO
'# @Version 1.00
'#####################################
      Error_Set = False
'### Error:
      str_Err                 = str_Err & "Error:                " & str_Error          & Chr(10)
'### Error-Code:
      str_Err                 = str_Err & "Code:                 " & str_ErrCode        & Chr(10)
'### Error-Line:
      str_Err                 = str_Err & "Line:                 " & str_Line           & Chr(10)
'### Current Function:
      str_Err                 = str_Err & "Function:             „ & str_Function       & Chr(10)
```

```
'### Current Function:
      str_Err                = str_Err & "Function:              "      & str_Function & Chr(10)
'### Calling LS-Function:
      str_Err                = str_Err & "called by:             "      & str_CalledBy & Chr(10)
'### Current Module:
      str_Err                = str_Err & "Module:                "      & str_Module   & Chr(10)
'### LS-Version:
      str_Err                = str_Err & "LS-Version:            "      & Lsi_info(6)  & Chr(10)
'### LS-Memory allocated:
      str_Err                = str_Err & "allocated Memory:      "      & Lsi_info(50) & Chr(10)
'### LS-Memory allocated from OS:
      str_Err                = str_Err & "allocated Memory OS:   "      & Lsi_info(51) & Chr(10)
'### LS-blocks used:
      str_Err                = str_Err & "blocks used:           "      & Lsi_info(52) & Chr(10)

'### Separator...
      str_Err                = str_Err & "================================" & Chr(10)
      Error_Set = True
End Function
```

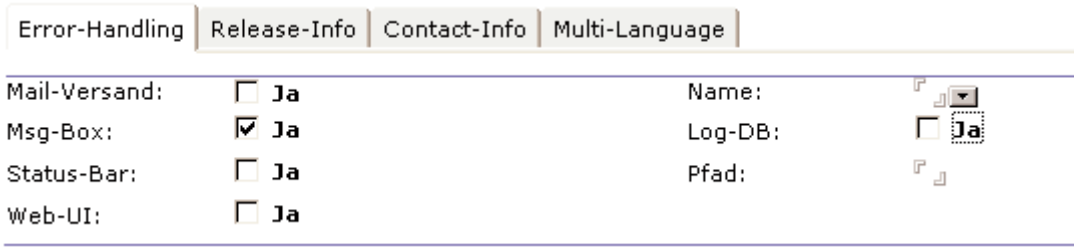
smartiX consulting gmbh

Code:	213
Line:	242
Function:	INITIALIZE
called by:	INITIALIZE
Module:	*8064EF4
LS-Version:	5.0.0.06
allocated Memory:	4041752
allocated Memory OS:	4369266
blocks used:	2414

==========

Error:	Path/file access error
Code:	75
Line:	243
Function:	INITIALIZE
called by:	INITIALIZE
Module:	*8064EF4
LS-Version:	5.0.0.06
allocated Memory:	4044192
allocated Memory OS:	4369266
blocks used:	2414

- Versand als Mail hilfreich aber nervend
- Loggen in Datenbank macht nur Sinn wenn diese Logs auch gelöscht werden (Performance)
- Loggen auf Server hilft nur wenn Zugriff besteht ;-)
- Information an den Anwender... Kann Irritationen hervorrufen
- Daher:



- Sind schnell
- Kaum dokumentiert
- Enthalten pro „Schluessel“ 1 Element
- Sind in der Lage alles nachzubauen
- Unheimlich flexibel
- Zwischenspeicher für wiederkehrende Abfragen (z. B. Reporting)
- Beliebig schachtelbar

- Type ist ein freier Datentyp

```
Type Employee
    FirstName as String
    LastName  as String
    Document  as NotesDocument
    Birthday  as NotesDateTime
    Others    List as Variant
End Type
```

- Type ist analog zu Klassen – ohne Konstruktor und ohne Methoden

```
Class Employee
    Private FirstName as String
    Private LastName  as String
    Private Document  as NotesDocument
    Private Birthday     as NotesDateTime
    Sub New(doc as NotesDocument)
       ... Your code...
    End Sub
    Public Property get getFirstName as string
        getFirstName  = Me.FirstName
    End Property
End Type
```

- Listen können aus verschiedenen Elementen bestehen

```
Dim li_FN    list as String
Li_FN(„Thomas“) = „Thomas“
```

- Oder

```
Dim li_FN    list as String
Li_FN(EmployeeID) = „Thomas“
```

- Diese Liste enthält mehrfach den Eintrag Thomas..

```
Dim li_FN    list as Employee
Dim Thomas as Employee
Li_FN(EmployeeID) = Thomas
```

- Dies sind alle Mitarbeiter in dem gewünschten Datenmodell der gew. Form

```
Dim li_FN    list as Employee
Dim Thomas as New Employee(doc_Curr)
Set Li_FN(EmployeeID) = Thomas
```

- Inhalte von Listen können wie folgt ausgelesen werden:

```
Msgbox(li_FN(„Thomas“))

EmployeeID        = „10283“
Msgbox(li_FN(„10283“))

EmployeeID        = „10283“
Msgbox(li_FN(„10283“).FirstName)

EmployeeID        = „10283“
Msgbox(li_FN(„10283“).getFirstName)
```

- Oder in Schlaufen durchlaufen werden...

```
Forall FirstName in li_FN
      Print FirstName & „ - „ & listtag(FirstName)
      ,### Thomas - Thomas
End Forall
```

```
Forall FirstName in li_FN
      Print FirstName & „ - „ & listtag(FirstName)
      ,### Thomas - 10283
End Forall
```

```
Forall Employee in li_FN
      Print Employee.FirstName & „ - „ & listtag(FirstName)
      ,### Thomas - 10283
End Forall
```

```
Forall Employee in li_FN
      Print Employee.getFirstName & „ - „ & listtag(FirstName)
      ,### Thomas - 10283
End Forall
```

- Oder aggregieren

```
If IsElement(li_FN(MyCompany)) then
    li_FN(MyCompany).Sum          = li_FN(MyCompany).Sum + NewSum
End if
```


- Erstellung einer Statistik für ein Bestellsystem welche folgende Anforderungen erfüllt:
  - Alle Bestellungen pro Lieferant und Monat
  - Anzahl der Bestellpositionen / Lieferant & Monat
  - Anzahl der Bestellungen / Lieferant & Monat
  - Wert der Bestellungen / Lieferant & Monat

- Listen können aus verschiedenen Elementen bestehen

```
Class Employee
    Private FirstName as String
    Private LastName  as String
    Private Document  as NotesDocument
    Private Birthday     as NotesDateTime
    Sub New(doc as NotesDocument)
       ... Your code...
    End Sub
    Public Property get getFirstName as string
        getFirstName  = Me.FirstName
    End Property
End Type
```

- Script-Funktion: Evaluate(„Code“, doc)
  - Wobei Code ein beliebiger Formelsprachen-Ausdruck sein kann
  - Und doc ggfls. ein Notesdocument ist gegen das die Formel evaluiert wird

- Liefert als Ergebnis ein „Array“ von Werten, z.B.

```
Dim va_Location  as Variant
Va_Location        = Evaluate(„@DBName“)
Msgbox(va_location(0))    ,### Servername
Msgbox(va_location(1))    ,### Dateipfad
```

- Zu früheren Zeiten nicht sooo stabil
- Unter Release 5 und 6 so unglaublich mächtig!!
- Früher wurde dies eingesetzt um „fehlende“ Script-Code bereitzustellen...

```
Dim va_Roles as Variant
Va_Roles          = Evaluate(„@UserRoles“)
Forall Role in va_Roles
    if Role       = „[Admin]“ then
    end if
End Forall
```

- Nächste Ebene ist die vereinfachte Handhabe von „Funktionalitäten“...

```
Dim va_Sort as Variant
Doc_Curr.fd_MyValues      = MyArray
Va_Sort                   = Evaluate(|@Sort(fd_dt_Plan;[ASCENDING])|, doc_Curr)
```

- Auf diese Weise erspart man sich jeden Sortier-Algorithmus ☺

- Wir wollen dem Anwender eine Auswahlliste geben...

```
Dim ses          As New NotesSession       '### We do not need this
Dim uiws         As New NotesUIWorkSpace
Dim va_Column    As Variant
Dim va_Select    As Variant
Dim an           As Integer
va_Column        = Evaluate(|@DBColumn("":"NoCache";@DBName;"va_all";1)|)
va_Select        = uiws.Prompt(6, "Select Value", "Please select a value
   or enter a new one", va_Column(0), va_Column)

an       = Msgbox("You have selected value: " &  va_Select)
```

- Und er soll uns einen Wert auswählen
  - Mit Fehler
  - Mit Fehlerhandling
  - Mit Mehrsprachigkeit !

- Wenn ich in der Lage bin....
  - Text als Code ausführen zu lassen...
  - Kann ich doch diesen Inhalt dynamisch aus Dokumenten auslesen...
  - Und mit dem identischen Code verschiedene Aktionen ausführen...

- Hierzu muss ich:
  - Code erstellen welcher auf definierte Elemente zugreift
  - Und kann dann in diesen Elementen alles hinterlegen was ich will...
  - Schnittstellen ???

- E-Procurement-System
- Zahlreiche Partner und diverse Anforderungen
  - XML
  - CSV
  - Formatierter Text
- Erhebliche Kosten bei Code-Anpassungen
- Wunsch möglichst viele Partner zu integrieren
- Einheitliche Code-Basis

- CSV

Links / Paths | Import | Creditor / Catalog | Owner | Mail-Confirmation | Mail-Order | 3.-Bezug | Mail-Approval | Mail-Rejection

Subject: "Bestellung von " + @GetProfileField("fa_contact";"fd_CompanyName_internal") + " von " + @Text(@Today)

EDI: (•) Ja ( ) Nein

Path WEB: c:\

Path Notes: c:\

FileName: @Text(fd_OrderNumber + ".asc")

Format Header:

```
"11" + "  " + "0701" + "00022736" + @Repeat("X";6) + @Repeat(" ";1) + @Repeat(" ";20) + @Left(fd_OrderNumber +
@Repeat(" ";30);30) + @Left("ALTANA PHARMA AG" + @Repeat(" ";30);30) + @Left(fd_OrderName + @Repeat(" ";30);30) +
@Left(@Trim(@Left(@Right(fd_OrderDeliv;" - ");" - ")) + @Repeat(" ";30);30) + @Left(@Left(fd_OrderDeliv;" ") + @Repeat("
";5);5) + @Left(@Right(@Left(fd_OrderDeliv;" - ");" ") + @Repeat(" ";30);30)  + @Left("DE" + @Repeat(" ";3);3) +
(@Left(@Right("0" + @Text(@Day(@Today));2) + @Right("0" + @Text(@Month(@Today));2) + @Right("0" +
@Text(@Year(@Today));4) +  @Repeat(" ";8);8)) + @Repeat(" ";8) + @Left(@Right(@Right(fd_OrderDeliv;" - ");" - ") +
@Repeat(" ";60);60) + @Repeat(" ";60) + @Repeat(" ";24) + @Repeat(" ";20) + @Repeat(" ";60) + @Repeat(" ";15) +
@Repeat(" ";15) + @Repeat(" ";15) + @Repeat(" ";15) + @NewLine +
"20" + "  " + "0701" + "00022736" + @Left("Irgendein Text den ich noch aussuchen muss" + @Repeat(" ";60);60)
```

Format Body:

```
"15" + "  " + "0701" + "00022736" + @Repeat(" ";8) + @right(@Repeat("0";8) + @Text(fd_ArtQuantity*100);8) +
@Repeat("0";8) + " " + @Repeat(" ";7) + @Repeat(" ";7) + @Left(fd_ArtNr + @Repeat(" ";7);7) + @Repeat(" ";30) +
@Repeat(" ";8) + " " + @Repeat(" ";15) + @Repeat(" ";15) + @Repeat(" ";15) + @Repeat(" ";15) + @Repeat(" ";15)
```

Format Footer:

- XML

Links / Paths | Import | Creditor / Catalog | Owner | Mail-Confirmation | Mail-Order | 3.-Bezug | Mail-Approval | Mail-Rejection

Subject: "Bestellung von " + @GetProfileField("fa_contact";"fd_CompanyName_internal") + " von " + @Text(@Today)

EDI: (•) Ja ( ) Nein

Path WEB: c:\

Path Notes: c:\

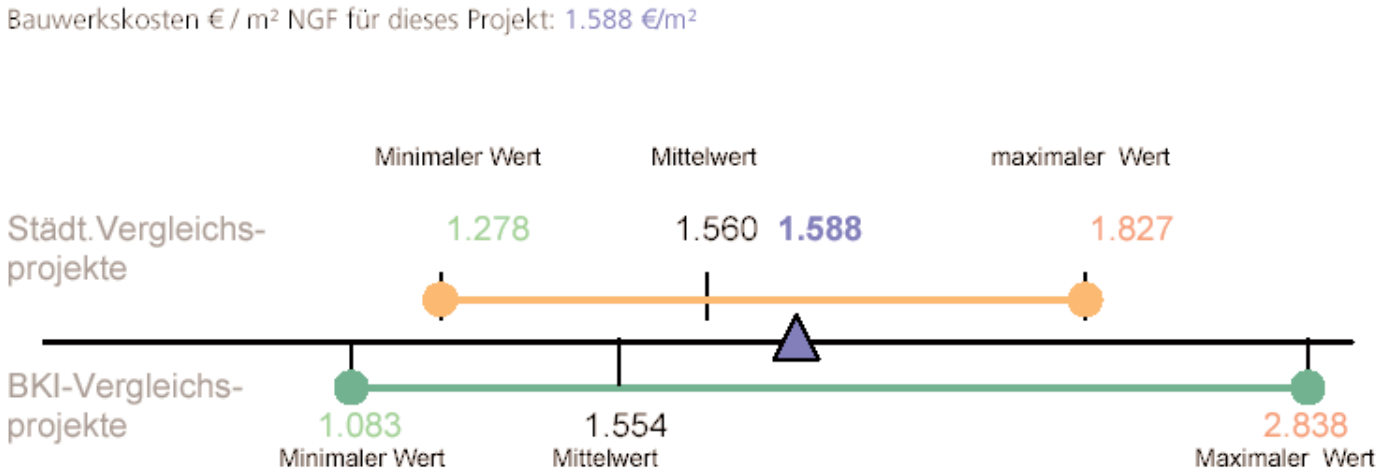
FileName: @Text(fd_OrderNumber + ".xml")

Format Header:

```
_A:="<?xml version = \"1.0\" encoding = \"ISO-8859-1\" ?>" +
"<ORDER type = \"release\"><ORDER_HEADER><ORDER_INFO>" +
"<ORDER_ID>" + fd_OrderNumber +"</ORDER_ID>"
+
"<ORDER_DATE>" + @Text(@Year(@Today)) + "-" + @Right("0" + @Text(@Month(@Today));2) + "-" + @Right("0" +
@Text(@Day(@Today));2) + "T" + @Text(@Time(@Now)) + "+01:00</ORDER_DATE>" +
"<ORDER_PARTIES>" +
"<BUYER_PARTY>" +

"<PARTY>" +
"<PARTY_ID type = \"supplier_specific\">" +
@Left(@Right(@GetProfileField("fa_shopdb";"fd_Additional_1");@Trim(fd_MyCompany) + "~");"~") + "</PARTY_ID>" +
"<ADDRESS>" +
"<NAME>" + fd_MyCompany + "</NAME>" +
"<DEPARTMENT>" + fd_OrderDep + "</DEPARTMENT>" +

"<CONTACT>" +
"<CONTACT_NAME>" + @Name([CN];fd_OrderName) + "</CONTACT_NAME>" +
"<PHONE type = \"office\">" + fd_OrderPhone + "</PHONE>" +
"<FAX>" + fd_OrderFax + "</FAX>" +
"<EMAIL>" + fd_OrderEmail + "</EMAIL>" +

"</CONTACT>" +
"<STREET></STREET>" +
"<ZIP></ZIP>" +
"<CITY></CITY>" +
"<COUNTRY>DE</COUNTRY>" +
"</ADDRESS>" +
```

• Anforderungen 1:

Auf Wunsch von Referat T soll ein Baukostenvergleich in das GA III aufgenommen werden. Da der wichtigste Wert beim Baukostenvergleich für wirtschaftliches Bauen sind die Bauwerkskosten in €/m² NGF. Diese Werte sollten grafisch wie folgt visualisiert werden:



Aufgabe:
Aus den Werten der u. g. Tabelle ist eine Grafik zu bilden für die Bauwerkskosten in Euro/m² NGF. Es ist eine Grafik (wie oben) zu generieren und in Notes zu importieren. Die Daten für diese Grafik bilden sich aus BKI (Baukostenindex) Daten (Minimalwert, Maximalwert, Mittelwert, Wert des konkreten Vorhabens).

- Anforderungen 2:

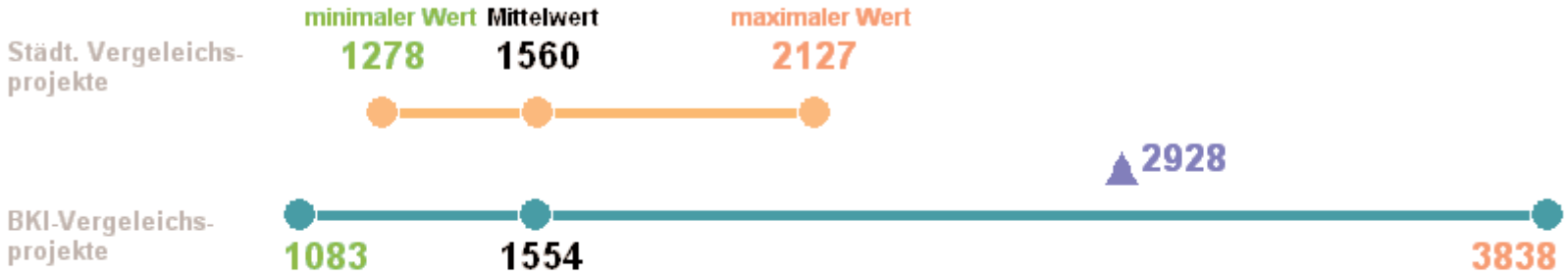
	Neubau Kindertagesstätten					
	Bauwerkskosten pro m^2 NGF			Gesamtbaukosten pro m^2 NGF		
	Minimal	Mittelwert	Maximal	Minimal	Mittelwert	Maximal
Dieses Projekt	**1588**			**2338**		
Städt. Vergleichsprojekte	1278	1560	1827	1886	2261	2663
BKI-Vergleichsprojekte *)	1083	1554	2838			
	Bauwerkskosten pro m^3 BRI			Gesamtbaukosten pro m^3 BRI		
Dieses Projekt	**402**			**592**		
Städt. Vergleichsprojekte	338	384	450	456	558	658
BKI-Vergleichsprojekte	168	340	517			
Bemerkungen *) Auf NGF umgerechnete BKI-Werte	Die Kosten dieses Projektes werden beeinflußt durch: * Barrierefreies Bauen (Aufzug) * Unterschreitung der Wärmeschutzverordnung * Regionale Baupreise (9,5 % über Bundesdurchschnitt)					

smartiX consulting gmbh

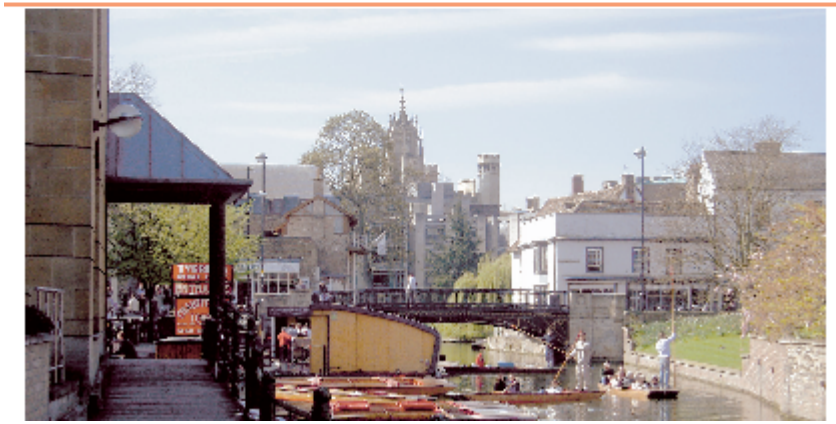
- Lösung 1:

Stadt Sektion Graphik

Bauwerkskosten €/m² NGF für dieses Projekt: 2928 €/m²



Stadt Sektion Bilder



EntwicklerCamp '06 Lotus Notes

B-KH

- Lösung 2:

	Bauwerkskosten pro m² NGF			Gesamtbaukosten pro m² NGF		
	Minimal	Mittelwert	Maximal	Minimal	Mittelwert	Maximal
Dieses Projekt		**fd_BWKTP** #			**fd_GBKTP** #	
Städt. Vergleichsprojekte		fd_BWKMin #			fd_GBKMin #	
		fd_BWKAv #			fd_GBKAv #	
		fd_BWKMax #			fd_GBKMax #	
BKI-Vergleichsprojekte		fd_BKIMin #				
		fd_BKIAv #				
		fd_BKIMax #				

```
<Table style="border-width:1px;border-color:#FF0000;padding:2px;margin:3px;border-style:solid;">
    <TR>
        <TD style="font-family:Verdana;font-size:10px;font-weight:bold;">
            Bauwerkskosten €/m<sup>2</sup> NGF für dieses Projekt: <Font color="blue"><Computed
Value> €/m<sup>2</sup> </Font><BR><BR>
        </TD>
    </TR>
    <TR>
        <TD>
<Computed Value>
        </TD>
    </TR>
```

Objects | Reference

- fd_Project (Field)
- fd_BWKTP (Field)
- fd_GBKTP (Field)
- fd_BWKMin (Field)
- fd_BWKAv (Field)
- fd_BWKMax (Field)
- fd_GBKMin (Field)
- fd_GBKAv (Field)
- fd_GBKMax (Field)
- fd_BKIMin (Field)
- fd_BKIAv (Field)
- fd_BKIMax (Field)
- fd_rt_MyRichText (Field)
- Database-link (Action)
- View-link (Action)
- Document-links (Action)
- Document-links (Action)
- Send document (Action)
- Module.BaseFields (Subform
- Module.Person (Subform)
- Computed Text
- Computed Text

Computed Text : Value

Run | Client | Formula

```
_MIN:=@Min(fd_BWKMIN:fd_BKIMIN);
_MAX:=@Max(fd_BWKMax:fd_BKIMax);
_TOTAL:=_MAX - _MIN;
_WIDTH:=600;
_RATIO:=_WIDTH/_TOTAL;
_LINKBLANK:="/" + @Subset(@DbName;-1) + "/bg_LinksTrans";
_GC:="/" + @Subset(@DbName;-1) + "/GreenCircle";
_OC:="/" + @Subset(@DbName;-1) + "/OrangeCircle";
_O:="/" + @Subset(@DbName;-1) + "/orange";
_G:="/" + @Subset(@DbName;-1) + "/green";
_TRIA:="/" + @Subset(@DbName;-1) + "/BlueTriangle";

_A:=@Text(_TOTAL) + " OO" +
@Text(@Min(fd_BWKMIN:fd_BKIMIN)) + " - " + @Text(@Max(fd_BWKMax:fd_BKIMax));

_BLText1:="<img src=\"" + _LINKBLANK + "\" width=\"" + @Text((_RATIO*(fd_BWKMin-_MIN))-30) + "px\" height=\"5px\"
```

- Script-Funktion: Execute(„Code“)
  - Wobei Code LotusScript-Code ist
  - Code wird abgearbeitet – ohne Debug-Option
  - Brilliante Option wenn man Code „nicht“ in der gewohnten Form aktualisieren will

- Herausforderung:
  - Übergabe von Objekten (Document) ist nicht möglich
  - Code läuft völlig dumm...
  - Keine dynamische „Einbindung“ möglich
  - Rückgabe von Objekten nahezu unmöglich

```
str_I%= Execute(str_C)
```

- Einfaches Beispiel

```
Dim ses        As New NotesSession'### We do not need
Dim str_Code As String       '### The code to execute
Dim i_Exec     As Integer    '### To execute the code...

str_Code       = |
                        dim MySes as New NotesSession
                        msgbox(MySes.UserName)
|

i_Exec = Execute(str_Code)
```

- Zugriff auf Dokument (einfach)

```
Dim ses         As New NotesSession   '### We do not need
Dim str_Code    As String         '### The code to execute
Dim i_Exec      As Integer        '### To execute the code...

str_Code        = |
                        dim MySes       as New NotesSession
                        dim doc_E       as NotesDocument
                        Set doc_E       = MySes.DocumentContext
                        msgbox(doc_E.UniversalID)
|

i_Exec  = Execute(str_Code)
```

- Rückgabe über „globale Variablen“

```
Dim ses    As New NotesSession          '### We do not need
Dim str_Code        As String  '### The code to execute
Dim i_Exec          As Integer           '### To execute the code...
str_Code  = |
            dim MySes                   as New NotesSession
            dim doc_E                                 as NotesDocument
            Set doc_E                   = MySes.DocumentContext
            str_MyUNID                  = doc_E.UniversalID
            msgbox(doc_E.UniversalID)
            Set doc_MyDoc       = doc_E
|

i_Exec      = Execute(str_Code)

Msgbox("And here is the same element once again..." & str_MyUNID)
```

- Übergabe an LS in Execute... (Schwierig...)

```
Dim ses  As New NotesSession       '### We do not need
Dim str_Code     As String         '### The code to execute
Dim i_Exec       As Integer        '### To execute the code...
Dim doc_Curr     as NotesDocument

Set doc_Curr     = ses.DocumentContext
str_Code         = |
          msgbox(doc_Curr.UniversalID)
|

i_Exec    = Execute(str_Code)

Msgbox("And here is the same element once again ..." & str_MyUNID)
```

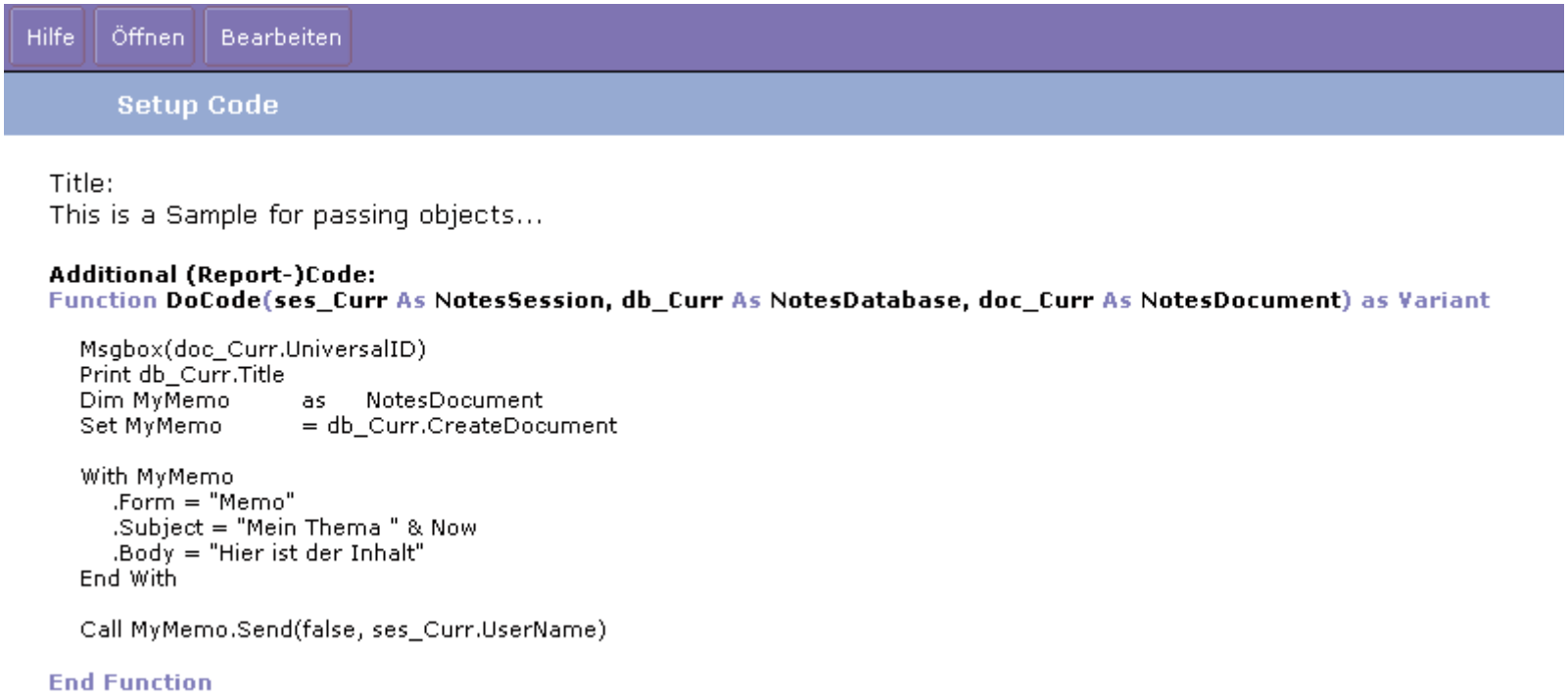
Funktioniert nur mit global definierten Variablen

- Globale Variablen können Probleme verursachen...
  - Verlust von Sessions / Objekten
  - Probleme bei der Klassen-Programmierung

- Alternativ könnte man auch einen kleinen Trick anwenden (komplexere Programmierung)

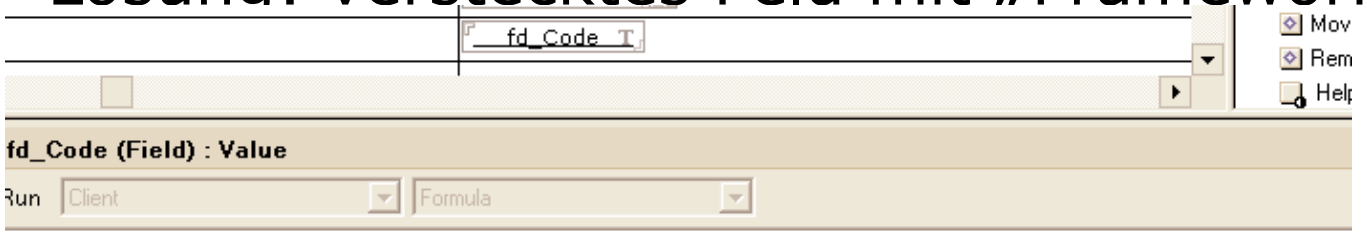
- Übergabe von „Konstanten“

- Das Ziel ist:



- Also eine Möglichkeit zu schaffen um einfach Script-Code auszuführen

- Lösung: verstecktes Feld mit „Framework“




```
_S:="Option Declare" + @NewLine +
"Sub Initialize" + @NewLine +
"Dim EvalSession as new NotesSession" + @NewLine +
"Dim db_Eval as NotesDatabase" + @NewLine +
"dim doc_HandOver as NotesDocument" + @NewLine +
"dim doc_Curr as NotesDocument" + @NewLine +

"dim str_MyName as string" + @NewLine +

"set db_Eval = EvalSession.CurrentDatabase" + @NewLine +
"set doc_Curr = db_Eval.GetDocumentByUNID(str_UNID)" + @NewLine +
"set doc_HandOver = db_Eval.CreateDocument" + @NewLine +
"doc_HandOver.Form = \"fa_mail\"" + @NewLine +
"Dim it_RT as New NotesRichTextItem(doc_HandOver, \"MyRT\")" + @NewLine+
"it_RT.Values = DoCode(EvalSession, db_Eval, doc_Curr)" + @NewLine +
"if it_RT.Values = \"\" then end" + @NewLine +
"str_MyName = Cstr(Cint(Rnd()*10000))" + @NewLine +
"doc_HandOver.fd_Name = str_MyName" + @NewLine +
"Call doc_HandOver.Save(true, true)" + @NewLine +
"End Cint(str_MyName)" + @NewLine +
"End sub" + @NewLine +
"function DoCode(ses_Curr as NotesSession, db_Curr as NotesDatabase, doc_Curr as NotesDocument) as Variant";
_E:="End Function";
@If(fd_CodeMan = "";"";
_s + @NewLine + fd_CodeMan + @NewLine + _E)
```

- In Kombination mit:

```
    Set ses             = New NotesSession
    Set db_Curr         = ses.CurrentDatabase
    Set dc_Curr         = db_Curr.UnprocessedDocuments
    If dc_Curr.Count > 0 Then
'### Get the document from the list....
        Set dc_Code  = uiws.PickListCollection(3, False, db_Curr.Server, db_Curr.Filepath, "va_runcode", "Auswahl Code", "Bitte wählen Si
        If dc_Code.Count > 0 Then
            Set doc_Code        = dc_Code.GetfirstDocument
            If doc_Code.fd_Code(0) = "" Then
            Else
                Set doc_Curr        = dc_Curr.GetFirstDocument
                While Not doc_Curr Is Nothing
                    Dim str_C As String
                    Dim str_I         As Integer
'### Make some adaptions to the code...
                    str_C = |Const str_UNID = "| & doc_Curr.UniversalID & |"| & Chr(10) & doc_Code.fd_Code(0)
                    str_I%= Execute(str_C)
                    Set doc_Curr      = dc_Curr.GetNextDocument(doc_Curr)
                Wend
            End If
        Else
        End If
    End If
```


smarti consulting gmbh

smartiX consulting gmbh
Emilienstr. 23
D-70563 Stuttgart
fon: +49 (0)711-459 991 30
fax: +49 (0)711-459 991 35
web: http://www.smartix.de

>> end