SWR

Hands-On Session

Dirk Alicke
Südwestrundfunk Stuttgart
www.swr.de

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

Datenaustausch zwischen Notes und Excel, Word und PowerPoint unter Nutzung von OLE-Automation

-

eine komplette Anleitung für Einsteiger

SWR

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

Was erwartet Sie?

- Voraussetzungen für den Datenaustausch
- Wirtschaftliche Gesichtspunkte
- Übersicht über Schnittstellen
- VBA-Code nach LotusScript übernehmen
- Excel, Word, PowerPoint
  Export, Import
- RichText-Felder (Ansätze zur Behandlung)
- Konstanten in MS-Applikationen

SWR

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

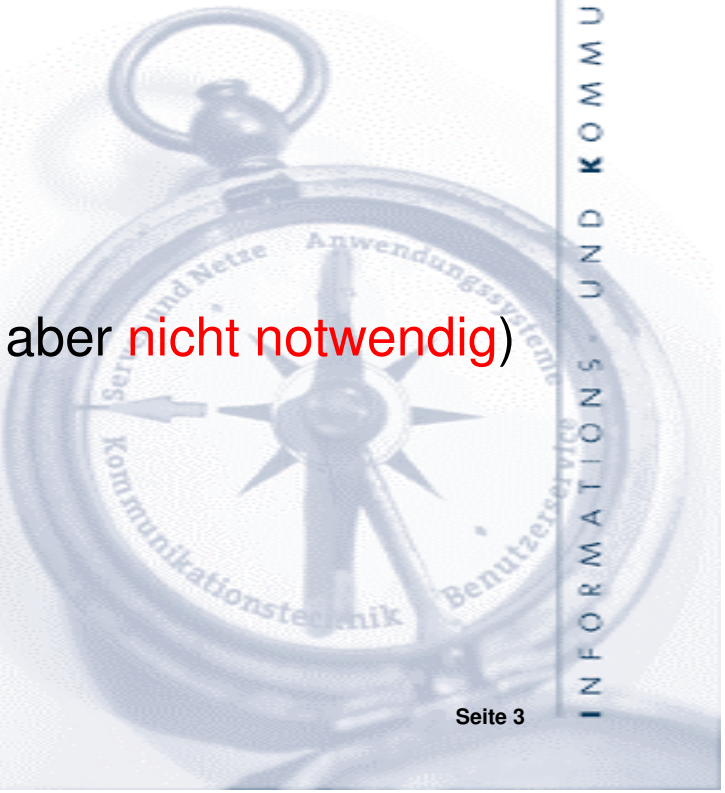
Software und Kenntnisse

Software

- Lotus Notes ab Version 4.x
- MS-Word
- MS-Excel
- MS-PowerPoint

Kenntnisse

- LotusScript
- Vorkenntnisse in VBA (vorteilhaft aber nicht notwendig)



Wirtschaftliche Gesichtspunkte

Beispiel

Die Sekretärin Schreibeschnell erstellt jeden Monat in Excel ein Diagramm. Dafür muss sie händisch aus 10 Anwendungen Daten übertragen und aufbereiten. Pro Anwendung benötigt sie 2 Stunden und die Personalkosten betragen pro Stunde 60,- • . Durch ein Script könnte diese Aufgabe auf einen Zeitaufwand von insgesamt 10 Minuten reduziert werden. Der Entwicklungsaufwand beträgt dabei 9900 • .

Wirtschaftliche Gesichtspunkte

Aufwand vor der Anpassung

$$A_{alt} = 10\frac{Anw}{Mon} * 2\frac{h}{Anw} * 60\frac{\bullet}{h}$$

$$A_{alt} = \underline{1.000\frac{\bullet}{Mon}}$$

Aufwand nach der Anpassung

$$A_{neu} = \frac{1}{6}\frac{h}{Mon} * 60\frac{\bullet}{h}$$

$$A_{neu} = \underline{10\frac{\bullet}{Mon}}$$

Einsparung pro Monat

$$E_{Mon} = 1.000\frac{\bullet}{Mon} - 10\frac{\bullet}{Mon}$$

$$E_{Mon} = \underline{990\frac{\bullet}{Mon}}$$

Return of Invest

$$R = \frac{9.900\bullet}{990\frac{\bullet}{Mon}}$$

$$R = \underline{\underline{10Mon}}$$

Schnittstellen

- **API … „Application Programming Interface“ (Schnittstelle zur Anwendungsprogrammierung)**
  - Schnittstelle, die ein Betriebssystem oder ein Softwaresystem zur Verfügung stellt
  - DLL … Dynamic Link Libraries

- **DDE … „Dynamic Data Exchange“**
  - Kanal zu einem DDE-fähigen Programm öffnen und Befehle übergeben
  - langsam und kompliziert
  - Programme verhalten sich nicht immer „richtig“ (Twainschnittstellenproblem)
  - beide Programme laufen sichtbar nebeneinander her

Was ist OLE?

- **OLE … „Object linking and / or embedding“**
  - sichtbar und unsichtbar
  - Zugriffslogik gegenüber DDE geändert
  - verlinken … Verweis auf das Objekt
  - einbetten … Kopie des Originals
  - Server-Anwendung
  - Objekte, Methoden und Eigenschaften stehen zur Verfügung

OLE – Versionen

- **OLE 1.0 und OLE 2.0**
  - OLE 1.0 startet das gesamte Programm
  - OLE 2.0 startet nur den Kernel (ca. 60 – 80% der Software)
  - kein „echtes“ Fernsteuern
- **OLE - Automation**
  - echtes Fernsteuern
- **OA – Office Automation**
  - anwendungsübergreifende Programmierung innerhalb von MS Office
  - neuer Begriff für OLE-Automation

Nachteile von OLE

- **Nachteile von OLE**
  - ist abhängig von den verwendeten Programmversionen (Softwareupdates und Sprachänderungen können Probleme bereiten)
  - Server-Anwendung muss installiert sein

Registry

- **HKEY_CLASSES_ROOT**
  - Programme, die per OLE angesprochen werden können
    z.B. Excel: Excel.Application
    CLSID ist ein weltweit eindeutiger Schlüssel
    {00024500-0000-0000-C000-000000000046}

- **HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID**
  - LocalServer32 … Excel.EXE mit Parameter /Automation
  - Nicht das komplette Excel wird gestartet, sondern nur der Funktionsteil (ca. 80%).

Demo: Registry

CreateObject

CreateObject

- neue Instanz einer Anwendung generieren
- Anwendung als Objekt zur Verfügung stellen
- Syntax: Set Object = CreateObject(ClassName)

Beispiele

```
Set wdApp = CreateObject("Word.Application")
Set xlApp = CreateObject("Excel.Application")
Set frApp = CreateObject("FineReader.Application")
Set ppApp = CreateObject("PowerPoint.Application")
Set ieApp = CreateObject("InternetExplorer.Application")
Set olApp = CreateObject("Outlook.Application")
Set acApp = CreateObject("Access.Application")
```

GetObject

GetObject

- Anwendung steht als Objekt zur Verfügung
- Anwendung anhand einer Datei starten
  (nur Parameter „PathName“ angeben)
- Prüfen, ob eine Anwendung gestartet ist bzw. gestartete Anwendung als Objekt zuweisen
  (nur Parameter „ClassName“ angeben)
- Syntax: Set Object = GetObject(PathName, ClassName)

Die erste OLE-Verbindung

```
‘ OLE-Verbindung zu Excel erzeugen
Set xlApp = CreateObject("Excel.application")
‘ existiert die OLE-Verbindung?
If xlApp is Nothing Then Exit Sub
xlApp.Visible = True                ‘ Excel anzeigen
‘ xlApp.Visible = False             ‘ Excel nicht anzeigen
xlApp.Workbooks.add                 ‘ neue Excel Mappe erstellen
```

```
‘ W I C H T I G, falls Visible auf False gesetzt wurde, muss die
Applikation zum Schluss beendet werden !
xlApp.Quit     ‘ Excel beenden
```

DEMO: erste OLE-Verbindung

SWR

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

1. Aufgabe

Untersuchen Sie das Verhalten beim Starten und Beenden der Anwendungen Word und Excel. Insbesondere die Punkte:
- Anwendung nach dem Start
- nach Setzen von Visible = True/False
- nach Quit
- nach ScriptEnde

Benutzen sie dabei den LotusScript-Debugger und den Windows Task-Manager.

Anmerkungen zu „Visible = True“

Die Anwendung wird sofort angezeigt (Visible = True)

+ **User sieht, dass sich etwas tut (Zappeleffekt).**
+ **Bei einem Fehler im Script kann die Anwendung vom User geschlossen werden (es bleiben keine Prozesse offen).**
- **User kann die Kommunikation stören (fehleranfälliger).**
- **Kommunikation zwischen den Anwendungen ist langsamer.**

Anmerkungen zu „Visible = False“

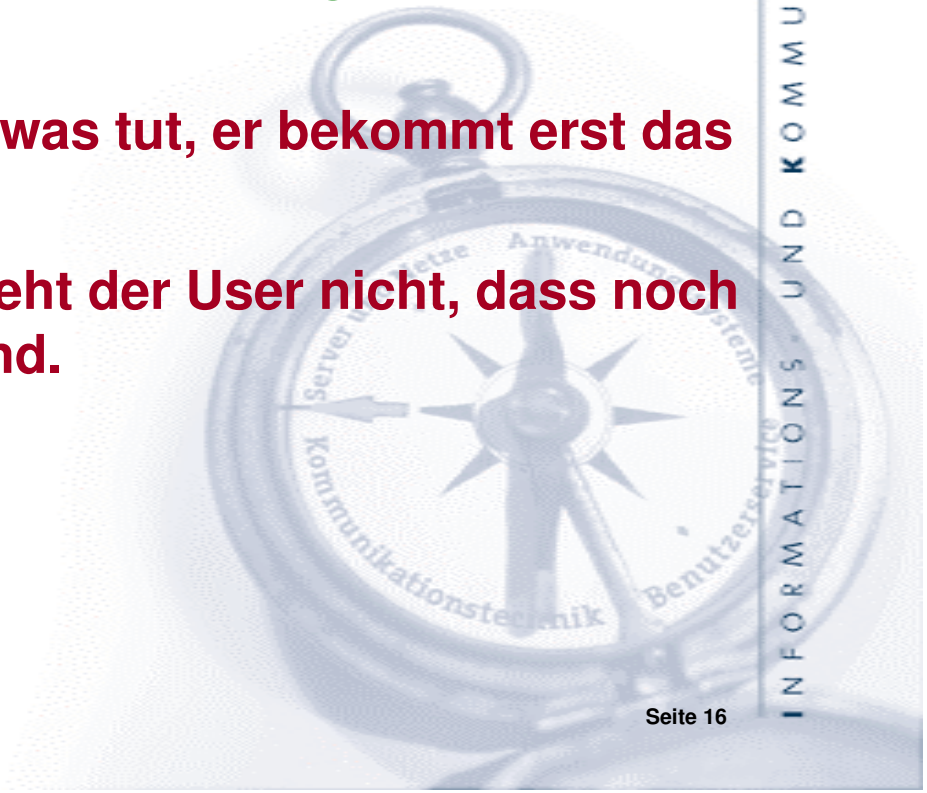
Die Anwendung wird nicht angezeigt (Visible = False):

+ User kann die Kommunikation nicht stören.

+ Kommunikation zwischen den Anwendungen ist schneller.

- User sieht nicht, dass sich etwas tut, er bekommt erst das Ergebnis zu Gesicht.

- Bei einem Fehler im Script sieht der User nicht, dass noch versteckte Prozesse offen sind.



Vorhandene Office-Makros verwenden

Word starten und Makro ausführen

Sinnvoll, wenn das Makro schon vorhanden ist.
Syntax: wdApp.Application.Run "Makro01 "

Vorteile

- User muss nicht mehrere Aktionen ausführen
- wenig Code im Notes
- schnelle Umsetzung, wenn das Makro vorhanden ist

Nachteile

- User braucht die Vorlage mit dem Makro
- Fehler müssen im Makro behandelt werden

SWR

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

2. Aufgabe

Zeichnen Sie ein Makro im Word auf.

Schreiben Sie einen Agenten, der die Anwendung Word öffnet und dieses Makro startet.

Makrorecorder Teil 1

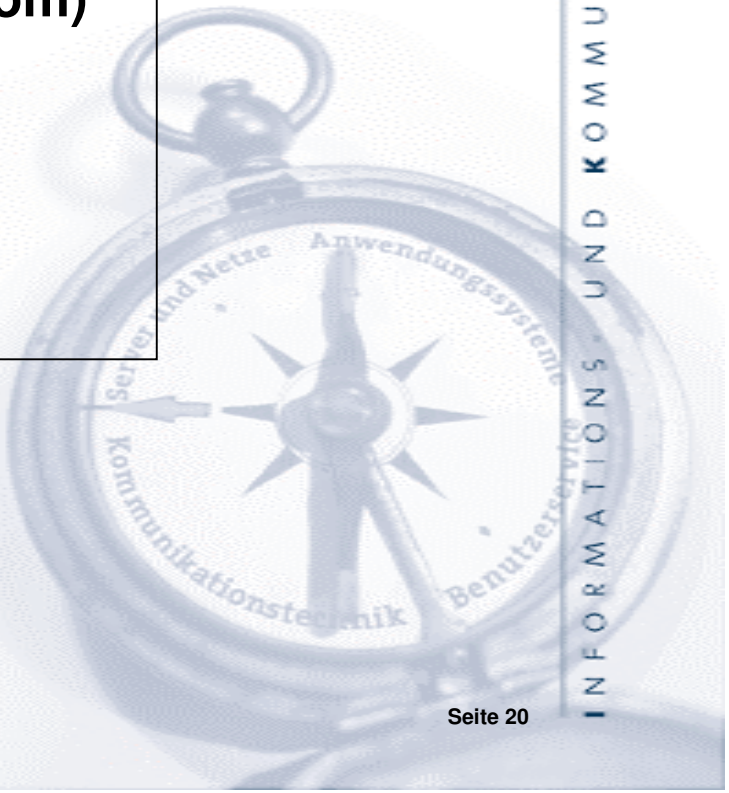
1. Makro mit Recorder aufzeichnen

```
Sub Makro1()
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    Selection.Borders(xlEdgeRight) = xlNone
End Sub
```

Makrorecorder Teil 2

2. Ballast abwerfen

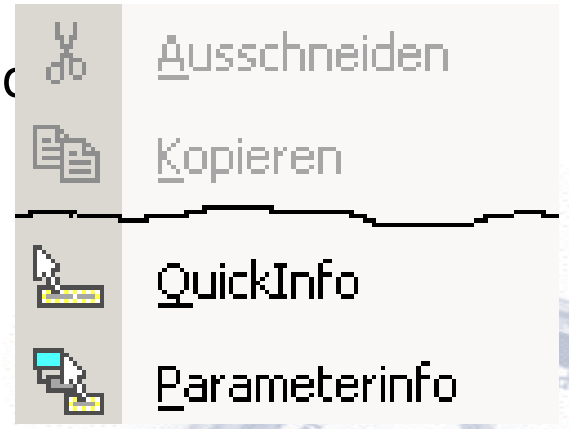
```
Sub Makro1()
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
End Sub
```



Makrorecorder Teil 3

3. Konstanten mit Quickinfo anzeigen lassen (rechte Maustaste auf Konstantennamen)

```
Sub Makro1()
    With Selection.Borders(xlEdgeBo
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
End Sub
```




```
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlConti
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
```

xlEdgeBottom = 9

Makrorecorder Teil 4

4. VBA-Konstanten ersetzen

```
Sub Makro1()
        With Selection.Borders(9)
                .LineStyle = 1
                .Weight = 2
                .ColorIndex = -4105
        End With
End Sub
```

5. Code nach Notes kopieren und syntaktisch anpassen

```
xlApp.Range("A1:J20").Select
With xlApp.Selection.Borders(9)
    .LineStyle = 1
    .Weight = 2
    .ColorIndex = -4105
End With
```

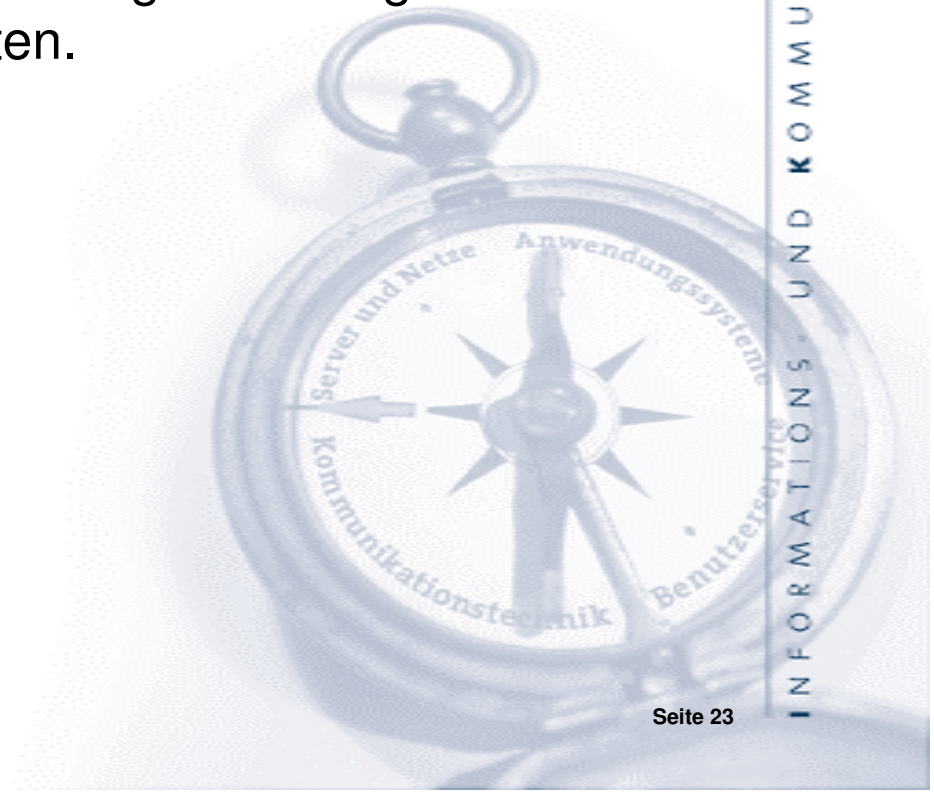
SWR

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

3. Aufgabe

Übernehmen Sie den Code des in Aufgabe 2 aufgezeichneten Makros in den LotusScript Agenten.



Konstanten

(Access 97, Excel 97, 5 und 7, Office 97, Office Binder 97, Outlook 97, VBA , Word 97)

Konstanten stehen in der VBA-Hilfe unter Konstanten

(Options)

```
%INCLUDE "ConstAccess97.lss"          ' 597 Const
%INCLUDE "ConstExcel97.lss"           ' 1266 Const
%INCLUDE "ConstOffice97.lss"          ' 794 Const
%INCLUDE "ConstOfficeBinder97.lss"    ' 11 Const
%INCLUDE "ConstOutlook2000.lss"       ' 251 Const
%INCLUDE "ConstPowerPoint2002.lss"    ' 816 Const
%INCLUDE "ConstVB.lss"                ' 279 Const
%INCLUDE "ConstVBA.lss"               ' 246 Const
%INCLUDE "ConstWord2000.lss"          ' 2395 Const
```

Excel-Start in eine Funktion auslagern

```
Function Excel_Start(xlApp As Variant, Anzeigen As Variant) As
Variant
‘ Anzeigen ... True oder False … soll Excel angezeigt werden?
    Excel_Start = False
    Set xlApp = Nothing
    Set xlApp = CreateObject("Excel.application")

    ‘ existiert die OLE-Verbindung
    If xlApp is Nothing Then Exit Function
    xlApp.Visible = anzeigen
    xlApp.Workbooks.add
    Excel_Start = True
End Function
```

Excelzellen lesen und schreiben

Schreiben

```
xlApp.Cells( Zeilennummer, Spaltennummer).Value = Wert
```

Beispiel:

```
For iZeile = 1 To 10
   For iSpalte = 1 To 20
     xlApp.Cells( iZeile,iSpalte).Value = iZeile * iSpalte
  Next
Next
```

Lesen

```
Value = xlapp.Cells(ZeilenNummer, SpaltenNummer).Value
```

Spaltenzahl in –buchstabe konvertieren

```
Function xlSZ2SB(Spaltenzahl As Long) As String
        If Fix((Spaltenzahl - 1) / 26) = 0 Then
                SP1 = ""
        Else
                SP1 = Chr(Fix((Spaltenzahl - 1) / 26) + 64)
        End If
        If Spaltenzahl Mod 26 = 0 Then
                SP2 = "Z"
        Else
                SP2 = Chr(Spaltenzahl Mod 26 + 64)
        End If
        xlSZ2SB = SP1 & SP2
End Function
```

Test: SpaltenZahlen2SpaltenBuchstaben

SWR

IKS

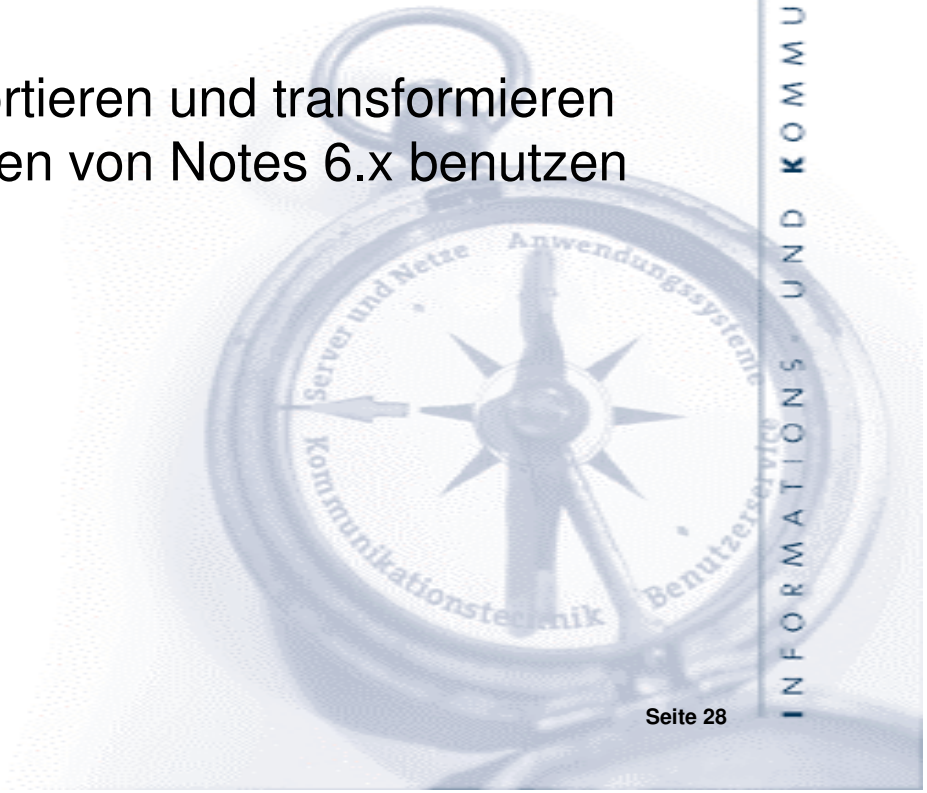
Ansätze für Richtext-Felder

Frontend-Dokument

- im Editmode per Zwischenablage
  Demo: Word Export (RTF)

Backend-Dokument

- eventuell als XML exportieren und transformieren
- eventuell Richtextklassen von Notes 6.x benutzen



INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

4. und 5. Aufgabe

Diagrammerstellung

Erstellen Sie eine Datenreihe mit Einnahmen und Ausgaben und erzeugen Sie ein Diagramm, welches den Trend der Einnahmen und Ausgaben ersichtlich macht.
Verwenden Sie für die Datenreihe Zufallszahlen bis 20.

Demo: Diagramm

Ansichten nach Excel exportieren

Erstellen Sie einen Agenten, der jede beliebige Ansicht nach Excel exportiert.
Testen Sie das Script an den Ansichten Ihres lokalen Adressbuches.

Demo: Ansichtsexport

6. und 7. Aufgabe

Export nach Word

Exportieren Sie die Dokumente aus der Ansicht „6. Aufgabe“ nach Word. Erstellen Sie dafür im Word eine A4-Seite im Querformat mit 2 Spalten.

Demo: Word Querformat

Export nach PowerPoint

Erstellen Sie eine Maske mit den Text-Feldern „Titel“ und „Body“ und erzeugen Sie mit dieser Maske einige Dokumente. Schreiben Sie jetzt einen Agent, der aus diesen Dokumenten einfache PowerPoint-Folien erstellt.

Demo: Export nach PowerPoint

8. Aufgabe

Export nach Excel

Importieren Sie aus der Excel-Datei sport.xls wahlweise die Einträge aus der Tabelle „Formel 1“ oder „Fußball WM 2006“.
Beachten Sie dabei die Feldformate (Datum, Zeit, Text).

Demo: Import Excel-Datei

SWR

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

Quellen

Hilfen

VBA
Notes

WEB-Links

msdn.microsoft.com/vbasic/
www.martinscott.com/dominosupersearch2.nsf/Search?OpenForm

Foren

www.dominoforum.de
www.atnotes.de

SWR

IKS

INFORMATIONS- UND KOMMUNIKATIONSSYSTEME

Vielen Dank für Ihre Aufmerksamkeit.

Jetzt sind Sie an der Reihe.