



DataFlex Entwickler Tag 2019

LibXL

Roman Köhler  
Gartenstadt-Genossenschaft Mannheim eG

# Anforderungen

- › Datenträgeraustausch mit anderen Unternehmen
- › Auswertungen aus dem Datenbestand zur anschließenden Weiterverarbeitung
- › Nachvollziehbare Kontrollen bei Umstellungen im laufenden System  
(Direkter Vergleich + Endsummen für Vorher / Nachher)

# MS-Office (Com)

- › Mächtig
- › Zu mächtig
  - › Wenn man nicht regelmäßig damit zu tun hat
  - › Wenn man sich nicht genau mit dem Objektmodell von Office auskennt
- › Suche nach einer einfacheren Lösung
  - › LibXL



# LibXL

- › Was ist LibXL
  - › Library für VDF
  - › Versionen
    - › Demoversion
      - › Hinweis in der 1 Zeile beim Schreiben
      - › Begrenzung der Zeilen beim Lesen
    - › Kaufversion

# LibXL

## › Vorteile

- › Kein Excel notwendig  
→ Unabhängig von unterschiedlichen COM-Versionen
- › Umfassende Sammlung von Methoden zur Erstellung, Lesen und Weiterverarbeitung von Excel-Tabellen
- › Format  
XLS + XLSX
- › Kurzer Einarbeitungszeit
- › Einfache Bedienung
- › Keine andere Programmiersprache oder –konzept (COM)

# LibXL

- › Vorteile
  - › Kurzer, übersichtlicher Quellcode
  - › Auch für “Unkundige” leicht zu lesen
  - › Einfach zu erweitern

# LibXL

- › Nachteile
  - › Nicht kostenlos

# LibXL

## › Inbetriebnahme

- › Quellcode in ein Unterverzeichnis kopieren
- › Library in der Workspace registrieren
- › Registrierdaten eingeben (eigene SubClass)

```
01 Use LibXL.pkg
02
03 Class cMyLibXL is a cLibXL
04
05     Procedure Construct_Object
06         Forward Send Construct_Object
07
08         Set psLicenseName to " "
09         Set psLicenseKey  to " "
10     End_Procedure
11 End_Class
```



# LibXL

- › Beispiele
  - › XLS lesen
  - › XLS schreiben
    - › Unformatierte Zellen
    - › Formatierte Zellen
    - › Eigene SubClass
    - › Formeln
    - › Mehrere Sheets
    - › Bilder

# LibXL

## › Beispiele

- › Alle Beispiele können auf der Homepage von Data Access heruntergeladen werden
- › LibXL separat herunterladen
- › Einschränkung:
  - › Keine Registrierung  
→ Kompilierbar, aber nur als Demoversion lauffähig

# LibXL-Beispiele

- › Beispiel 1: Tabelle lesen
  - › Tabelle öffnen (BookHandle vergeben)  
Arbeitsblatt öffnen (SheetHandle vergeben)
  - › Dimensionen ermitteln
  - › Tabelle Zelle für Zelle durchgehen und die Werte ausgeben
  - › Handles löschen

# LibXL-Beispiele

## Beispiel 1: Tabelle lesen

```
// Objektdefinition
Object oLibXL is a cMyLibXL
End_Object

// Tabelle öffnen
Get Load of oLibXL sFileName to hoBook
Get Sheet of hoBook 0 to hoSheet
Get LastRow of hoSheet to iRows
Get LastCol of hoSheet to iCols
// Tabelle durchsuchen
For iRow from (FirstRow(hoSheet)) to iRows
    For iCol from (FirstCol(hoSheet)) to iCols
        Get ReadValue of hoSheet iRow iCol to sValue
    Loop
Loop

// Handle löschen
Send Destroy of hoBook
```



# LibXL-Beispiele

- › Beispiel 2: Tabelle lesen (Array)
  - › Erweiterung von Beispiel 1
    - › Zwischenspeichern der ermittelten Daten in einem Array
    - › Ausgabe in einem cjGrid

# LibXL-Beispiele

- › Beispiel 3: Tabelle schreiben (unformatiert)
  - › Erweiterung von Beispiel 2
    - › Speichern der ermittelten Werte in einer neuen Excel-Tabelle
    - › Werte sind noch unformatiert

# LibXL-Beispiele

## Beispiel 3: Tabelle schreiben

```
// Handle erstellen & Sheet hinzufügen
Get CreateXMLBook of oLibXL to hoBook
Get AddSheet of hoBook "Demo (unformatiert)" to hoSheet

// Spaltenbreite setzen
Set piColumnWidth of hoSheet iCol to iWidth

// Zelle füllen
Send WriteString of hoSheet iRow iCol sValue

// Datei speichern
Send SaveAs of hoBook sFileName

// Handle löschen
Send Destroy of hoBook
```

# LibXL-Beispiele

- › Beispiel 4: Tabelle schreiben (formatiert)
  - › Erweiterung von Beispiel 3
    - › Werte formatiert
      - › Schriftgröße
      - › Zahlenformatierung



# LibXL-Beispiele

## Beispiel 4: Tabelle schreiben (formatiert)

```
// Handle für Schriftarten
Get AddFont of hoBook sFontName nFontSize to hoFont10

// Handle für Format hinzufügen und Schriftart zuweisen
Get AddFormat of hoBook to hoTextFormat
Set phoFont of hoTextFormat to hoFont10

// Linksbündiges Textformat erstellen
Get Duplicate of hoTextFormat to hoTextFormatLeft
Set peAlignH of hoTextFormatLeft to XL_ALIGNH_LEFT
Set peAlignV of hoTextFormatLeft to XL_ALIGNV_CENTER

// Formatiertes Schreiben
Send WriteStringWithFormat of hoSheet iRow iCol sValue FormatHandle
Send WriteNumberWithFormat of hoSheet iRow iCol nValue FormatHandle

// Weitere Write-Commands:
WriteBoolean(WithFormat), WriteBlank(WithFormat), WriteFormula(WithFormat),
WriteFormulaNum(WithFormat), WriteFormulaStr(WithFormat),
WriteFormulaBool(WithFormat)
```

# LibXL-Beispiele

- › Beispiel 5: Tabelle schreiben (formatiert mit eigener Klasse)
  - › Erweiterung von Beispiel 4
    - › Verlagerung der Formatierungen aus Beispiel 4 in eine eigene Klasse
    - › → Wiederverwertbar

# LibXL-Beispiele

## Beispiel 5: Tabelle schreiben (eigene Klasse)

```
// Klasse erstellen, Handles definieren
Class cMyLibXLWithFormats is a cMyLibXL
    Procedure Construct_Object
        Property Handle phoHeaderFormatLeft

// Prozedur zur Initialisierung der Formate
Procedure InitializeMyLibXL Handle hoBook
    Get AddFont of hoBook "Arial" 10 to hoFont10

    // Handle für Format hinzufügen und Schriftart zuweisen
    Get AddFormat of hoBook to hoTextFormat
    Set phoFont of hoTextFormat to hoFont10

    // Linksbündiges Textformat erstellen
    Get Duplicate of hoTextFormat to hoTextFormatLeft
    Set peAlignH of hoTextFormatLeft to XL_ALIGNH_LEFT
    Set peAlignV of hoTextFormatLeft to XL_ALIGNV_CENTER

    // Properties belegen
    Set phoHeaderFormatLeft to hoHeaderFormatLeft
```

# LibXL-Beispiele

## Beispiel 5: Tabelle schreiben (eigene Klasse)

```
// Nummernformat zuweisen
Get Duplicate of hoTextFormat to ho0DezimalstellenFormat
Set peNumFormat of ho0DezimalstellenFormat to (AddCustomNumFormat(hoBook,
"###,##0"))
```

```
// Im Programm
// Objekt definieren und initialisieren
Get CreateXMLBook of oLibXL to hoBook
Send InitalizeMyLibXL of oLibXL hoBook
Send WriteStringWithFormat of hoSheet iRow iCol sValue FormatHandle
Send WriteNumberWithFormat of hoSheet iRow iCol nValue FormatHandle
```



# LibXL-Beispiele

- › Beispiel 6: Tabelle schreiben (formatiert mit eigener Klasse) + Formeln
  - › Erweiterung von Beispiel 5
    - › Die Summenzellen werden nun mit Formeln und nicht mit den eingelesenen Werten gefüllt

# LibXL-Beispiele

## Beispiel 6: Tabelle schreiben (... + Formeln)

```
// Formel mit Format schreiben
```

```
Send WriteFormulaWithFormat of hoSheet iRow iCol sFormula FormatHandle
```

# LibXL-Beispiele

- › Beispiel 7: Tabelle schreiben (weitere Sheets & Bilder)
  - › Hinzufügen weiterer Arbeitsblätter
  - › Einfügen von Bildern
    - › In Originalgröße
    - › In 2,5-facher Vergrößerung
    - › Ermittlung der im Arbeitsblatt enthaltenen Bilder

# LibXL-Beispiele

## Beispiel 7: Sheets & Bilder

```
// Sheet finden
Get SheetByName of hoBook "Demo 3" to hoSheet

// Anzahl der Bilder im Sheet
Get PictureCount of hoSheet to iNumPics

// Voraussetzung
xlPictureId iPictureID

// Bild in Originalgröße einfügen
Get AddPictureFromFile of hoSheet sPictureName to iPictureID
Send SetPicture of hoSheet iPictureID iRow iCol iOffsetX iOffsetY

// Bild in 2,5 facher Vergrößerung einfügen
Get AddPictureFromFile of hoSheet sPictureName to iPictureID
Send SetPictureScaled of hoSheet iPictureID iRow iCol iOffsetX iOffsetY nZoom
```



# LibXL-Beispiele

- › Extras
  - › XLS(X) öffnen / schreibenHinzufügen weiterer Arbeitsblätter
  - › Schriftattribute
    - › Color
    - › Bold
    - › Italic
    - › Strikeout
    - › Underline

# LibXL-Beispiele

## › XLS(x) öffnen

- › Der Lesevorgang ist für beide Dateitypen gleich

```
Get Load of oLibXL sFileName to hoBook
```

## › XLS(x) schreiben

- › Hier werden unterschiedliche Funktionen aufgerufen:

```
// XLS erstellen
```

```
Get CreateBook of oLibXL to hoBook
```

```
// XLSx erstellen
```

```
Get CreateXMLBook of oLibXL to hoBook
```

# LibXL-Beispiele

## › Schriftartenattribute

### › Schriftfarbe

```
Set piColor of hoFont to XL_COLOR_???  
// Definierte Farben in LibXL.h
```

### › Schriftart "Fett"

```
Set pbBold of hoFont to True
```

### › Schriftart "Kursiv"

```
Set pbItalic of hoFont to True
```

### › Schriftart "Durchgestrichen"

```
Set pbStrikeout of hoFont to True
```

### › Schriftart "Unterstrichen"

```
Set peUnderlineStyle of hoFont18 to XL_UNDERLINE_???  
// Definierte Underlines in LibXL.h
```

# LibXL-Beispiele

## › Zellenattribute:

### › Zeilenumbruch

`Set pbWrapText of hoTextFormat to True`

### › Rotation

`Set piRotation of hoTextFormat to iDegrees`

# LibXL-Beispiele

## › Zeilen-/Spaltenattribute

### › Zeilenhöhe

`Set piRowHeight of hoSheet to iRow rHeight`

### › Spaltenbreite

`Set piColumnWidth of hoSheet iCol to rWidth`

### › Zeilen gruppieren

`Send GroupRows of hoSheet iRowStart iRowEnd True`

### › Spalten gruppieren

`Send GroupColumns of hoSheet iColStart iColEnd True`



# LibXL-Beispiele

## › Drucken:

### › Papierformat

```
Set pePaper of hoSheet to XL_PAPER_???  
// Definierte Papiergrößen in LibXL.h
```

### › Hoch-/Querformat

```
Set pbLandscape of hoSheet to True
```

# LibXL-Beispiele

## › Drucken:

### › Zentrieren

`Set pbCenterHorizontally of hoSheet to True`

`Set pbCenterVertically of hoSheet to True`

### › Ränder

`Set prMarginLeft of hoSheet to (CMToInch(1.5))`

`Set prMarginRight of hoSheet to (CMToInch(1.6))`

`Set prMarginBottom of hoSheet to (CMToInch(1.7))`

`Set prMarginTop of hoSheet to (CMToInch(1.8))`

# LibXL-Beispiele

## › Drucken:

### › Kopf- / Fußzeile

```
Set psHeaderText of hoSheet to "&CHeaderText"  
Set psFooterText of hoSheet to;  
  ("&L&D  &T &C&Z" + "\" + "&F&R&P" + "/" + "&N")
```

// Auszug aus Code-Erklärungen: LibXL.pkg

&L / &C / &R = Beginn Left- / Center- / Right-Section

&D = Datum

&T = Zeit

&Z = Speicherort

&F = Dateiname

&P = Aktuelle Seite

&N = Gesamtzeilenanzahl

# LibXL - Unklarheiten

- › Zellenattribute:
  - › Rahmen
  - › Rahmentyp
  - › Rahmenfarbe
  - › Füllfarbe
- › Arbeitsmappenattribute:
  - › Gitter aktivieren/deaktivieren
  - › Arbeitsmappe sperren
  - › Arbeitsmappe mit Passwort sperren

# LibXL - Unklarheiten

- › Drucken
  - › Druckbereich festlegen
  - › An Seite anpassen
- › Lösungen für diese Probleme bitte im Beispielcode einfügen und an [Koehler@Gartenstadt-Genossenschaft.de](mailto:Koehler@Gartenstadt-Genossenschaft.de) schicken.  
Der Download wird dann entsprechend aktualisiert.



# LibXL

- › Hilfe?
  - › Minimaler Schnelleinstieg:  
Am Anfang der LibXL.pkg
  - › Referenz im Web:  
<http://libxl.com/documentation.html>
  - › [Forum Data Access Worldwide](#)

# LibXL

- › Preise
  - › 495,00 € / Programmierer
  - › 247,50 € Subscription
- › Bezugsquellen:
  - › Data Project GmbH  
<http://www.data-project.net>
  - › DataAccess Europe  
<https://www.dataaccess.eu>



LibXL





DataFlex Entwickler Tag 2019

Vielen Dank für Ihre Aufmerksamkeit!  
Haben Sie Fragen?