

Business Reports with L^AT_EX

Reporting, Controlling & Co.

Uwe Ziegenhagen, Cologne
ziegenhagen@gmail.com

9. Oktober 2012

Motivation

- ▶ My employer is a „Private Equity“ fund manager in Cologne, belongs to Deutsche Bank
- ▶ Investor letters & reportings plus regulatory reporting (Bafin, SEC, etc.)
- ▶ Workflow: proprietary LoB¹ software + SQL Server \Rightarrow Business Intelligence Layer \Rightarrow Excel \Rightarrow Word \Rightarrow PDF
- ▶ Excel \Rightarrow Word \Rightarrow PDF requires huge amount of time
- ▶ Proof of Concept:
 - ▶ How could a T_EX-based workflow look alike?
 - ▶ What can be automated?

¹„Line of Business“

My Setup

- ▶ Windows 7 64-bit with XAMPP (Apache, PHP, MySQL)
- ▶ phpMyAdmin & HeidiSQL to work with MySQL
- ▶ Python 2.7 with MySQLDB², Eclipse with Pydev³
- ▶ MySQL ODBC Connector 64-bit to get data into Excel
- ▶ Excel2L^AT_EX 2.1⁴

²<http://sourceforge.net/projects/mysql-python/>

³<http://pydev.org/>

⁴<http://www.ctan.org/tex-archive/support/excel2latex/>

Sample Data

- ▶ „Northwind“ Database, included in Access & SQL Server
- ▶ marakana.com/download/sql/northwind.sql
for MySQL, set to UTF8

Tables

- | | |
|------------------------|---------------|
| ▶ Categories | ▶ Orders |
| ▶ Customercustomerdemo | ▶ Products |
| ▶ Customerdemographics | ▶ Region |
| ▶ Customers | ▶ Shippers |
| ▶ Employees | ▶ Suppliers |
| ▶ Employeeterritories | ▶ Territories |
| ▶ Order_details | ▶ Usstates |

Some new tables and columns were added.

Desirable Reports & Documents

Management Information

- ▶ Volume of Sales by customer and employee
- ▶ Sales growth
- ▶ Outstanding invoices
- ▶ Outstanding orders
- ▶ Lists of inventory
- ▶ Financial overviews

Other useful information

- ▶ Generation of invoices
- ▶ General correspondence

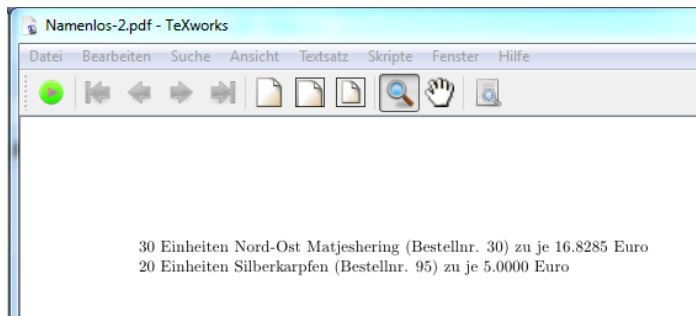
⇒ Premise: Keep the toolchain simple!

1st example: Generating a letter via SQL only

- ▶ UNION statement to mix separate SQL commands
- ▶ Number of columns must match

```
1 SELECT "\\documentclass{article}\\r\\n\\begin{document}"
2 UNION
3 SELECT "\\begin{itemize}"
4 UNION
5 SELECT Concat("\\item ", 2*ReorderLevel, " Einheiten ",
6 productname, " ( Bestellnr . ", productid, ") zu je ",
7 unitcost, " Euro" )
8 FROM products
9 WHERE discontinued = 0
10 AND unitsinstock < reorderlevel
11 AND unitsonorder = 0
12 UNION
13 SELECT "\\end{itemize}\\r\\n\\end{document}"
```

Result



Conclusion

- ▶ In general it works, but not very good style!
- ▶ Mixing $\text{T}_\text{E}\text{X}$ and SQL may become a mess. . .
- ▶ Change of $\text{T}_\text{E}\text{X}$ -code requires change of the query

Next step: MySQL & Python

- ▶ Have MySQL deliver the data (only), have Python & L^AT_EX do the rest
- ▶ „Rest“ means
 - ▶ Request the data from MySQL
 - ▶ Format the result
 - ▶ Create the L^AT_EX file
 - ▶ Control the compilation process

Python (Wikipedia)

- ▶ general-purpose, interpreted
- ▶ design philosophy \Rightarrow code readability
- ▶ syntax said to be clear and expressive
- ▶ large and comprehensive standard library

Python – Minimal Database Example

```
1  import MySQLdb
2  mysql_opts = {
3      'host': "localhost", 'user': "uwe",
4      'pass': "xxxxxxxxxx", 'db': "pydb" }
5
6  mysql = MySQLdb.connect(mysql_opts['host'],
7      mysql_opts['user'], mysql_opts['pass'],
8      mysql_opts['db'])
9
10 cursor = mysql.cursor()
11 cursor.execute("SELECT * FROM 'products'")
12 products = cursor.fetchall()
13
14 print products
```

Python – Writing L^AT_EX code

```
1 headerfile = open("f:/header.tex","r")
2 header = headerfile.read()
3 footerfile = open("f:/footer.tex","r")
4 footer = footerfile.read()
5
6 f = open("f:/test.tex", "w")
7 f.write(header)
8 f.write("Hallo \\LaTeX!\r\n");
9 f.write(footer)
10 f.close()
```

Compiling the source either with

- ▶ `os.system('pdflatex filename.tex')`
- ▶ or use `latexmk.py`
(<http://pypi.python.org/pypi/latexmk.py/>)

PythonT_EX

- ▶ by Geoffrey Poore
- ▶ current version: 0.9beta3 from 17.07.2012
- ▶ <https://github.com/gpoore/pythontex>
- ▶ various options for formatting the output
- ▶ supports scientific computing packages such as PyLab⁵, SymPy⁶

Installation

- ▶ Install Python (if not available on your machine)
- ▶ Download PythonT_EX from <https://github.com/gpoore/pythontex>
- ▶ Run the installer script for T_EX Live
- ▶ Use it!

⁵<http://www.scipy.org/PyLab>

⁶<http://code.google.com/p/sympy/>

Python – Writing L^AT_EX code

```
1 \documentclass{minimal}
2 \usepackage{pythontex}
3 \begin{document}
4 \begin{pyconsole}[] [frame=single]
5 x = 123
6 y = 345
7 z = x + y
8 z
9 def f(expr):
10     return(expr**4)
11
12 f(x)
13 print('Python says hi from the console!')
14 \end{pyconsole}
15 \end{document}
```

Workflow

```
>>> x = 123
>>> y = 345
>>> z = x + y
>>> z
468
>>> def f(expr):
...     return(expr**4)
...
>>> f(x)
228886641
>>> print('Python says hi from the console!')
Python says hi from the console!
```

- ▶ Run pdf_latex on the T_EX-code!
- ▶ Run pythontex2 / 3 on the T_EX-code!
- ▶ Run pdf_latex on the T_EX-code again⁷!

pycode executed, but not typeset

pyverb typeset, but not executed

pyblock typeset and executed

pyconsole interactive Python console

⁷see appendix for T_EXworks integration

Back to the first example. . .

```
1 \usepackage{pythontex}
2 \begin{document}
3 \begin{pycode}{}
4 import MySQLdb as mdb
5 import sys
6 con = mdb.connect('localhost', 'read', 'bSVTuFQWqAdBK5nA', 'northwind');
7 with con:
8     cur = con.cursor()
9     cur.execute("SELECT Concat(2*ReorderLevel,' Einheiten ',productname, ' (
        Bestellnr . ', productid, ') zu je ', round(unitcost,2), ' Euro' )
        FROM products WHERE discontinued = 0 AND unitsinstock < reorderlevel
        AND unitsonorder = 0")
10
11 rows = cur.fetchall ()
12 f = open("test2.tex", "w")
13 for row in rows:
14     f.write("\\item " + str(row[0])+"\\r")
15 f.close ()
16 \end{pycode}
17
18 \begin{itemize}
19 \input{./pythontex-files-orderlist2/test2}
20 \end{itemize}
```

Result

- 30 Einheiten Nord-Ost Matjeshering (Bestellnr. 30) zu je 16.83 Euro
- 20 Einheiten Silberkarpfen (Bestellnr. 95) zu je 5.00 Euro

Another Example – List of Open Payments

OrderID	Shipped on	Total
11072	15.05.1998	5218.0
11008	06.05.1998	4903.5
11039	15.05.2000	3090.0
11068	16.08.1998	2384.8
11070	05.09.1998	1873.5
11059	15.07.1998	1838.0
11077	11.07.1998	1374.6
11045	08.08.1998	1309.5
11076	29.09.1998	1057.0
11058	12.08.1998	858.0

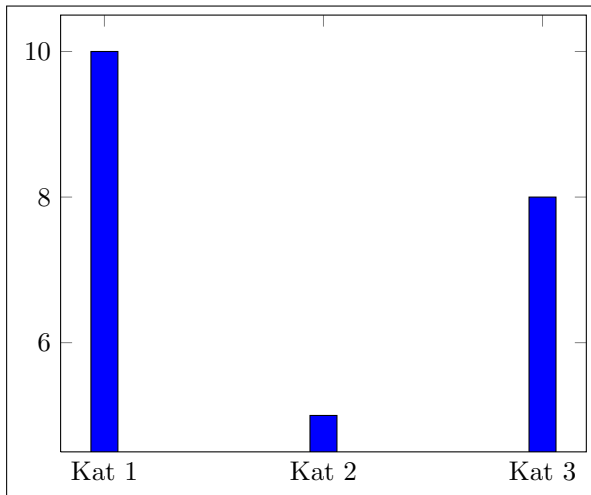
Source

Graphics with pgfplots

- ▶ written by Christian Feuersänger, based on PGF & TikZ
- ▶ >400 pages documentation

```
1 \documentclass{standalone}
2 \usepackage{pgfplots}
3 \begin{document}
4 \begin{tikzpicture}
5 \begin{axis}[
6     symbolic x coords={Kat 1,Kat 2,Kat 3},
7     xtick=data]
8     \addplot[ybar, fill =blue] coordinates {
9         (Kat 1,10)
10        (Kat 2,5)
11        (Kat 3,8)
12    };
13 \end{axis}
14 \end{tikzpicture}
15 \end{document}
```

Result



Using Python_T_EX with MySQL I

```
1 \documentclass{scrartcl}
2 \usepackage{pythontex}
3 \usepackage[landscape,a4paper, left =1cm,
4 right =1cm,top=2.5cm,bottom=1cm]{geometry}
5 \usepackage{pgfplots}
6 \pagestyle{empty}
7
8 \begin{document}
9
10 \begin{pycode}[]
11 import MySQLdb as mdb
12 import sys
13 import os
14
15 con = mdb.connect('localhost', 'read',
16                  'bSVTuFQWqAdBK5nA', 'northwind');
17
```

Using PythonT_EX with MySQL II

```
18 with con:
19     cur = con.cursor()
20     cur.execute("SELECT e.FirstName, \n\
21     round(SUM(det.unitprice * det.quantity),2) as total,\n\
22     o.EmployeeID\n\
23     FROM order_details det\n\
24     INNER JOIN orders o\n\
25     ON o.orderid = det.orderid\n\
26     inner join employees e\n\
27     on e.EmployeeID = o.EmployeeID\n\
28     group by o.EmployeeID\n\
29     order by 3")
30
31 rows = cur.fetchall ()
32
33 f = open("emps.dat", "w")
34 f.write("Employee\t Revenue\t xpos\n")
35
```

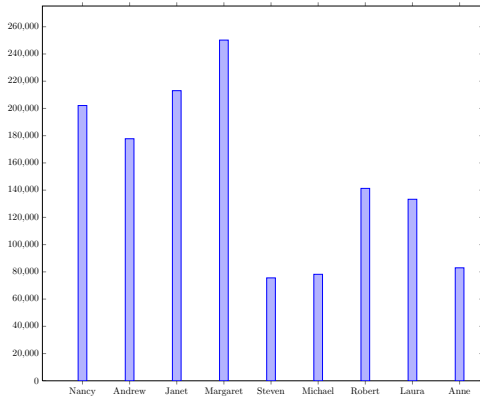
Using PythonT_EX with MySQL III

```
36 for row in rows:
37     f.write(" " + str(row[0]) + "\t" + str(row[1]) + "\t" + str(row
        [2]) + "\n")
38 f.close()
39 \end{pycode}
40
41 \centering
42 \begin{tikzpicture}
43 \begin{axis}[
44     ybar, xtick=data, ymin=0, xmax=9.5,
45     xticklabels from table={pythontex-files-empRevenuePyT/emp.dat
        }{Employee},
46     width=0.7\textwidth,
47     scaled y ticks = false,
48     yticklabel style={/pgf/number format/fixed}
49 ]
50
51
```

Using PythonT_EX with MySQL IV

```
52 \addplot table [  
53     x= xpos,  
54     y=Revenue,  
55     width= 15cm  
56 ] {pythontex-files-empRevenuePyT/emp.dat};  
57 \end{axis}  
58 \end{tikzpicture}  
59 \end{document}
```

Result



Conclusion

- ▶ Mixing \LaTeX + scripting language fairly easy
- ▶ Python \TeX combines \TeX 's power with Python's elegance
- ▶ Pgfplots is pretty powerful but needs some time to learn
- ▶ other interesting things:
 - ▶ PGF pie charts
(<http://code.google.com/p/pgf-pie/>)
 - ▶ Excel + \LaTeX
 - ▶ Excel + Powershell + \LaTeX

To be continued...

T_EXworks Integration

following <http://code.google.com/p/texworks/wiki/AdvancedTypesettingTools>

- ▶ Create a batch file „pythonize.bat“ in the T_EX live bin directory that contains (for Windows & Python 2.7):
- ▶ `pdflatex -synctex=1 "%1"&& pythontex2 "%1"&& pdflatex -synctex=1 "%1"`
- ▶ Create a new tool entry in the T_EXworks settings

