



## Das Apache POI-Framework als Reporting- Tool für Oracle Anwendungen

Referent:

**Sven-Olaf Kelbert,**

Fachbereichsleiter „Oracle Development“,

MT AG, Ratingen



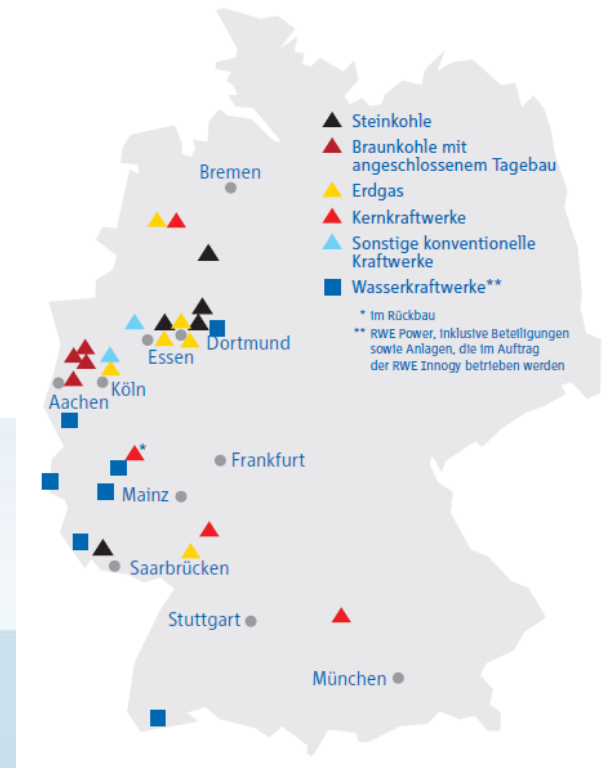


## MT AG MANAGING TECHNOLOGY – ENABLING THE ADAPTIVE ENTERPRISE

- Gründung 1994
- Inhabergeführte AG:  
Aktienkapital 1.500.000 €
- Hauptsitz Ratingen;  
Niederlassungen Dortmund, Frankfurt
- Mitarbeiter:  
> 200 Festangestellte  
> 65 Freie Mitarbeiter
- Full-Service-Dienstleistung für alle  
Phasen des Software-Lifecycle
- Herstellerunabhängige Expertise in  
den marktführenden Technologien wie  
Oracle, IBM, Microsoft, SAP und  
OpenSource
- Themen- und Lösungs-Know-how in den  
Kerndisziplinen des Adaptive Enterprise

- Ausgangslage
- fachlicher Hintergrund
- technischer Hintergrund
- POI-Framework
- Umsetzung
- Probleme und Lösungen
- Vergleich alt ↔ neu
- Ausblick

- unter den 5 führenden Strom- und Gaserzeugern in Europa
- 70.000 Mitarbeiter
- 53 Mrd. € Umsatz



- eines der führenden Unternehmen der Energiegewinnung und -erzeugung in Deutschland



- Organisationseinheit Strombezug im Bereich Vertragsmanagement
  - Abwicklung von Fremdstrombezugsrechnungen verschiedener Kraftwerke.
  - Unterstützung: seit 2002 System IAIS (Integriertes Abrechnungs- und Informationssystem)

- basierte auf der Entwicklungsplattform Microsoft Visual Basic 6 (VB6)
  - seitens Microsoft Unterstützung eingestellt
  - Möglichkeit von zukünftigen Erweiterungen eingeschränkt
  - Komponenten von Drittherstellern, die nicht für die neuen Oracle-Datenbanken zertifiziert sind

→ Modernisierung der Applikation unumgänglich

- IT-Dienstleister für die RWE Power
- analysierte die Rahmenbedingungen
- leitete eine Neu-Entwicklung der Anwendung in die Wege
  - Unterstützung durch MT AG
  
- Entwicklungswerkzeug: Oracle Forms 10g
  - schon in vielen anderen Konzern-Anwendungen im Einsatz
  - Möglichkeit, Synergie-Effekte zu nutzen

- komplexe Excel-Schnittstelle wesentlicher Bestandteil der Altanwendung
    - durch die technischen Möglichkeiten von VB6 sehr eng mit dem System verwoben
    - enge Verknüpfung von IAIS und Excel unter Oracle Forms nicht mehr möglich
- Schnittstelle musste nahezu komplett neu entwickelt werden.
- Nutzung des Apache POI-Framework
    - erfolgreich bei vielen Oracle Forms Projekten der RWE



- Ausgangslage
- fachlicher Hintergrund
- technischer Hintergrund
- POI-Framework
- Umsetzung
- Probleme und Lösungen
- Vergleich alt ↔ neu
- Ausblick

- Abwicklung und Prognose von Strombezügen und -lieferungen verschiedener Kraftwerke.
- Pflege der Geschäftspartner und deren Verträge mit der RWE Power
- Analyse von Strommengendaten verschiedener Stromzähler (sog. Zählpunkte) an den Kraftwerken.
  - viertelstündliche Ermittlung und Import über verschiedene Schnittstellen nach IAIS
  - Ausgangsbasis für die Berechnung von Monatsverbräuchen, Anzahl von Kraftwerksstillständen und Wiederanfahrungen der Kraftwerke

- komplexe Berechnungskomponenten mit speziellen Formeln
- Werte der verschiedenen Zählpunkte gegeneinander verrechnen
  - verschiedenen Preiskomponenten und Tarifen
  - Prüfung von Rechnungen bzw. Gutschriften der Geschäftspartner
- Darstellung der Ergebnisse der Berechnungskomponenten sowie kompletter Zeitreihen von Zählpunkten in Form von Auswertungen und Controlling-Berichten in Excel
- Unterstützung des internen Berichtswesens
  
- Berechnungen von vergangenen Monaten
- Prognosen in die Zukunft

- Ausgangslage
- fachlicher Hintergrund
- **technischer Hintergrund**
- POI-Framework
- Umsetzung
- Probleme und Lösungen
- Vergleich alt ↔ neu
- Ausblick

- Import der Viertelstundenwerte der Strommengen über verschiedene Schnittstellen nach IAIS
  - in zwei Tabellen je Monat jeweils etwa 4 Mio. neue Datensätze
  - je Tabelle momentan ca. 350 Mio. Datensätze
- Monatsanfang: Berechnung sog. Ausgabewerte aus den Werten der Zählpunkte des Vormonats mit Hilfe von Berechnungskomponenten
- Darstellung der Ergebnisse dieser Ausgabewerte in Excel
  - Kontrolle der vorliegenden Rechnungen
- diverse in der Datenbank gespeicherte Excel-Vorlagen für jeden Geschäftspartner und jeden Vertrag

- enthält verschiedene Tabellenblätter
  - Ergebnisse der Ausgabewerte
  - Liste aller Viertelstundenwerte eines Zählpunktes
  - Liste aller Betriebszeiten und Ausfallzeiten eines Kraftwerks
  - Listen der Strom- und CO2-Preise der Leipziger Strombörse
- wird im Rahmen der Rechnungsprüfung mit den benötigten Daten aus der Datenbank verknüpft, um die gewünschte Auswertung zu erzeugen.
- änderbar
- Zeitraum flexibel

	A	B	C	D
1				MONAT_1
2	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_1	
3	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_2	
4	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_3	
5	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_4	
6	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_5	
7	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_6	
8	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_7	
9	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_8	
10	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_9	
11	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_10	
12	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_11	
13	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_12	
14	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_13	
15	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_14	
16	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_15	
17	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_16	
18	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_17	
19	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_18	

- 7 fest definierte Tabellenblätter in jeder Auswertung

Zwischenablage				Schriftart				Ausrichtung				Zahl				Formatvorlagen				Zellen			
Geschäftspartner_XY.Kraftwerk_Z.Ausgabewert_1_MONAT_1				fx				13707570															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O								
1				August 2010	September 2010	Oktober 2010	November 2010	Dezember 2010	Januar 2011	Februar 2011	März 2011	April 2011	Mai 2011	Juni 2011	Juli 2011								
2	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_1	13707570	19798250	38206280	25863870	25849360	26799930	30382817,8	22431600	22278292,5	16988012,97	17057466	<Fehler>								
3	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_2	13707570	19798250	38206280	25863870	25849360	26799930	30382817,8	22431600	22278292,5	16988012,97	17057466	<Fehler>								
4	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_3	5	5	5	5	5	5	5	5	5	5	5	5								
5	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_4	114492800	109111870	127051330	110742790	123003470	116725870	110159200	124011470	121506920	99520320	93601060	<Fehler>								
6	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_5	0	0	0	0	0	0	0	0	0	0	0	<Fehler>								
7	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_6	0,05	0,05	0,05	0,05	0,05	0,03	0,03	0,03	0,03	0,03	0,03	0,03								
8	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_7	0,044	0,044	0,044	0,044	0,044	0,044	0,044	0,044	0,044	0,044	0,044	0,044								
9	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_8	0,074	0,074	0,074	0,074	0,074	0,074	0,074	0,074	0,074	0,074	0,074	0,074								
10	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_9	0,056	0,056	0,056	0,056	0,056	0,059	0,059	0,059	0,059	0,059	0,059	0,059								
11	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_10	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,92	0,92								
12	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_11	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07								
13	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_12	2963	2963	2963	2963	2963	2907	2907	2907	2907	2907	2907	2907								
14	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_13	0,134	0,134	0,134	0,134	0,134	0,147	0,147	0,147	0,147	0,147	0,147	0,147								
15	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_14	19,9	19,9	19,9	19,9	19,9	20,85	20,85	20,85	20,85	20,85	20,85	20,85								
16	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_15	0,803	0,803	0,803	0,803	0,803	0,846	0,846	0,846	0,846	0,846	0,846	0,846								
17	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_16	3,16	3,16	3,16	3,16	3,16	3,38	3,38	3,38	3,38	3,38	3,38	3,38								
18	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_17	3875	3875	3875	3875	3875	3803	3803	3803	3803	3803	3803	3803								
19	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_18	2963	2963	2963	2963	2963	2907	2907	2907	2907	2907	2907	2907								
20	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_19	2035	2035	2035	2035	2035	1997	1997	1997	1997	1997	1997	1997								
21	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_20	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07	42695,07								
22	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_21	1,19	1,19	1,19	1,19	1,19	1,19	1,19	1,19	1,19	1,19	1,19	1,19								
23	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_22	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Zeitreihe>	<Fehler>								
24	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_23	0	0	0	0	0	0	0	0	0	0	0	0								
25	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_24	-4735690	-3622670	-1912860	-6733810	-7576080	-3744570	-3349612,8	-1361511,2	-4146602,1	-4834604,4	-3776908,4	<Fehler>								
26																							

- Möglichkeit, individuelle Tabellenblätter zu definieren,
- grafische oder tabellarische Aufbereitung der Daten fürs Controlling

	August 2010	September 2010	Oktober 2010	November 2010	Dezember 2010	Januar 2011	Februar 2011	März 2011	April 2011	Mai 2011	Juni 2011
	kW	kW	kW	kW	kW	kW	kW	kW	kW	kW	kW
	0	0	0	0	0	0	0	0	0	0	0
Einzelpreis	3,160	3,160	3,160	3,160	3,160	3,380	3,380	3,380	3,380	3,380	3,380
Zwischenwert Preis	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
alter Preis	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
neuer Preis	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Gesamtpreis	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	kWh	kWh	kWh	kWh	kWh	kWh	kWh	kWh	kWh	kWh	kWh
	0	1.420	0	0	0	0	9.110	0	0	24.157	0
Einzelpreis	0,00803	0,00803	0,00803	0,00803	0,00803	0,00846	0,00846	0,00846	0,00846	0,00846	0,00846
Gesamtpreis	0,00	11,40	0,00	0,00	0,00	0,00	77,07	0,00	0,00	204,37	0,00
Gesamtpreis	0,00	11,40	0,00	0,00	0,00	0,00	77,07	0,00	0,00	204,37	0,00
Messung:	0,50%	0,50%	0,50%	0,50%	0,50%	0,50%	0,50%	0,50%	0,50%	0,50%	0,50%
Gesamtpreis	0,00	11,40	0,00	0,00	0,00	0,00	77,07	0,00	0,00	204,37	0,00
Gesamtpreis	0,00	-0,06	0,00	0,00	0,00	0,00	-0,39	0,00	0,00	-1,02	0,00
tzanlagen:											
Gesamtpreis	42.695,07	42.695,07	42.695,07	42.695,07	42.695,07	42.695,07	42.695,07	42.695,07	42.695,07	42.695,07	42.695,07
	kvarh	kvarh	kvarh	kvarh	kvarh	kvarh	kvarh	kvarh	kvarh	kvarh	kvarh
	13.707.570	19.798.250	38.206.280	25.863.870	25.849.360	26.799.930	30.382.818	22.431.600	22.278.293	16.988.013	17.057.466
Einzelpreis	0,00920	0,00920	0,00920	0,00920	0,00920	0,00920	0,00920	0,00920	0,00920	0,00920	0,00920
Gesamtpreis	126.109,64	182.143,90	351.497,78	237.947,60	237.814,11	246.559,36	279.521,92	206.370,72	204.960,29	156.289,72	156.928,69
	3.875,00	3.875,00	3.875,00	3.875,00	3.875,00	3.803,00	3.803,00	3.803,00	3.803,00	3.803,00	3.803,00
	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00	5,00
Gesamtpreis	1.614,58	1.614,58	1.614,58	1.614,58	1.614,58	1.584,58	1.584,58	1.584,58	1.584,58	1.584,58	1.584,58
	2.035,00	2.035,00	2.035,00	2.035,00	2.035,00	1.997,00	1.997,00	1.997,00	1.997,00	1.997,00	1.997,00
Simulation	2.035,00	2.035,00	2.035,00	2.035,00	2.035,00	1.997,00	1.997,00	1.997,00	1.997,00	1.997,00	1.997,00



- Ausgangslage
- fachlicher Hintergrund
- technischer Hintergrund
- **POI-Framework**
- Umsetzung
- Probleme und Lösungen
- Vergleich alt ↔ neu
- Ausblick

- Java-APIs zum Erstellen und Bearbeiten von Microsoft-Office-Dateien
  - Excel
  - Word
- POI = „Poor Obfuscation Implementation“ (schlechte, verschleiende Implementierung)
  - nicht das POI-Framework selbst gemeint, sondern das Microsoft-Office-Dateiformat
- mehrere Komponenten
  - HSSF für Excel



**The Apache POI Project**

<http://poi.apache.org>

Klasse	Funktionalität
HSSFWorkbook	repräsentiert eine Excel-Datei
HSSFSheet	repräsentiert ein Tabellenblatt
HSSFRow	repräsentiert eine Zeile eines Tabellenblatts
HSSFCell	repräsentiert eine Zelle einer Zeile
HSSFCellStyle	repräsentiert das Format einer Zelle
HSSFName	repräsentiert den Namen eines Zellbereichs

Methode	Funktionalität
HSSFSheet createSheet (String sheetName)	erstellt ein neues Tabellenblatt
removeSheetAt (int index)	löscht ein Tabellenblatt
int getNumberOfSheets()	gibt die Anzahl der Tabellenblätter zurück
String getSheetName (int index)	gibt den Namen des Tabellenblattes zurück
setSelectedTab (int index)	selektiert ein Tabellenblatt
HSSFName createName()	definiert einen neuen Zellbereich
removeName (int index)	entfernt einen Zellbereich
write (FileOutputStream fos)	schreibt die Excel-Datei ins Filesystem

<b>Methode aus HSSFSheet</b>	<b>Funktionalität</b>
boolean getProtect()	gibt an, ob das Tabellenblatt geschützt ist
setProtect (boolean protection)	schützt das Tabellenblatt bzw. gibt es frei
HSSFRow createRow (int rownum)	erstellt eine neue Zeile
removeRow (HSSFRow row)	löscht eine Zeile

<b>Methode aus HSSFRow</b>	<b>Funktionalität</b>
HSSFCell createCell(short column)	Erstellt eine neue Zelle

## Methode aus HSSFCell

## Funktionalität

HSSFCellStyle getCellStyle()

gibt das Aussehen der Zelle zurück

setCellStyle(HSSFCellStyle style)

setzt das Aussehen der Zelle

setCellValue(String wert)

schreibt einen Wert in eine Zelle

setCellFormula(String formula)

schreibt eine Formel in eine Zelle

## Methode aus HSSFCellStyle

## Funktionalität

setDataFormat(short format)

setzt das Format der Zelle

## Methode aus HSSFName

## Funktionalität

setNameName(String wert)

setzt den Namen eines Zellbereichs

setReference(String cellArea)

definiert das Ausmaß des Zellbereichs

# Möglichkeiten von POI

	A	B	C	D	E	F
1	Nicht formatiert				<b>Weiß - Schwarz</b>	
2	Standard formatiert				<b>YELLOW - DARK_BLUE</b>	
3	<b>Fett formatiert</b>				<b>DARK_RED - GOLD</b>	
4	<i>Kursiv formatiert</i>				Red Border - GOLD	
5	durchgestrichen					
6	<u>Unterstrich einzeln</u>					
7	<u>Unterstrich doppelt</u>					
8						
9	Arial 8	Courier Ne	<i>Monotype Corsiva 8</i>			
10	<b>Arial 12</b>	Courie	<i>Monotype Corsiva 12</i>			
11	<b>Arial 16</b>	Couri	<i>Monotype Corsiva 16</i>			
12						
13	Type Offset 2					
14	Type Offset -2					
15						

- Ausgangslage
- fachlicher Hintergrund
- technischer Hintergrund
- POI-Framework
- **Umsetzung**
- Probleme und Lösungen
- Vergleich alt ↔ neu
- Ausblick



- Java-Klasse als Wrapper für die POI-Funktionalitäten

```
public int schreibeNumber(double wert) {
    int result = RETURN_OK;
    if (m_cell != null) {
        try {
            if (m_cell.getCellType() != HSSFCell.CELL_TYPE_NUMERIC) {
                setDataformat(STYLE_STANDARD);
            }
            m_cell.setCellValue(wert);
            result = RETURN_OK;
        } catch (Exception e) {
            simpleErrors(e, "schreibeNumber");
            result = RETURN_FEHLER;
        }
    } else {
        result = RETURN_ZELL_NICHT_SELEKTIERT;
    }
    return result;
}
```

- Einbindung der neuen Java-Klasse in Forms über den Java-Importer
- Methoden als generiertes PL/SQL-Package

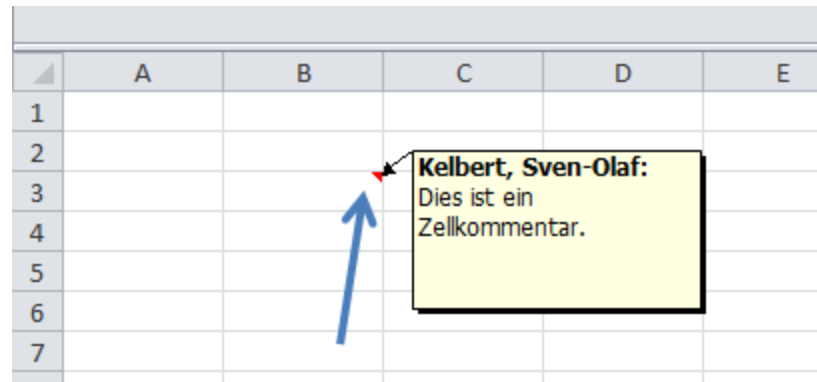
```
FUNCTION schreibeNumber(  
  obj   ORA_JAVA.OBJECT,  
  a0    NUMBER) RETURN NUMBER IS  
BEGIN  
  args := JNI.CREATE_ARG_LIST(1);  
  JNI.ADD_DOUBLE_ARG(args, a0);  
  RETURN JNI.CALL_INT_METHOD(FALSE,  
                               obj,  
                               'com/rwe/iais/excel/IaisExport',  
                               'schreibeNumber',  
                               '(D)I',  
                               args);  
END;
```

- Berechnung sämtlicher Ausgabewerte, die in irgendeiner Form in der Excel-Datei erscheinen mussten, mit ihren zugrunde liegenden Formeln in der Datenbank
- Ergebnisse (Zahlen, Texte oder Zeitreihen) → Object Type in der Datenbank
  - Struktur wie die Excel-Datei, unterschiedliche Strukturen mit Untertypen für die einzelnen Reiter
- Laden der zur Auswertung passenden Excel-Vorlage aus der Datenbank auf den Application Server per WebUtil
- Schreiben der Ergebnisse der Auswertung auf den jeweils passenden Reiter mit den POI-Funktionalitäten.
- Datei speichern und per Webutil auf den Client kopieren

- weitere spezielle Excel-Datei mit Makro aus der Datenbank auf den Client kopieren.
- Excel-Datei mit dem Makro auf dem Client starten.
- Makro öffnete dann die eigentliche Auswertungs-Datei,
  - führt Nacharbeiten durch und wird anschließend wieder gelöscht.
- Auswertungs-Datei blieb offen für Anpassungen oder Ergänzungen
  - nur auf den individuellen Reitern
  - 7 fest vorgegebene Reiter per Blattschutz gegen Änderungen geschützt.

- Ausgangslage
- fachlicher Hintergrund
- technischer Hintergrund
- POI-Framework
- Umsetzung
- Probleme und Lösungen
- Vergleich alt ↔ neu
- Ausblick

- Einsatz der POI-Version 2.5 aufgrund von Kompatibilitätsvorgaben
  - aktuell: 3.7
- Bug: keine Verarbeitung von Zellkommentaren und grafischen Elementen wie Rechtecken oder Linien
  - Excel-Vorlagen wurden korrupt und ließen sich nicht mehr öffnen
  - Elemente in den Auswertungen nicht zwingend notwendig, konnten entfernt werden



- Keine Möglichkeit in POI 2.5, Spalten auf ihre optimale Breite zu setzen.
  - notwendig, weil die 7 festen Tabellenblätter mit Blattschutz geschützt werden
  - Spaltenbreite nicht manuell von den Anwendern änderbar.
- Umweg über Visual Basic Makro
  - spezielle Excel-Vorlage mit Makro aus der Datenbank geladen.
  - Dateiname der eigentlichen Auswertung in die Datei
  - auf dem Client Makro-Datei geöffnet.
  - Makro setzt optimale Breite einer jeden Spalte
  - Makro-Datei wird gelöscht, so dass nur noch die eigentliche Auswertung vorhanden ist.

- alle Ergebnis-Zellen des ersten Standard-Reiters sind mit einem Zellnamen versehen,
- einfachere Referenz auf den individuellen Reitern

Zwischenablage		Schriftart		Ausrichtung			
Geschäftspartner_XY.Kraftwerk_Z.Ausgabewert_1_MONAT_1				<i>fx</i>	13707570		
	A	B	C	D	E	F	
1				August 2010	September 2010	Oktober 2010	Nov
2	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_1	13707570	19798250	38206280	
3	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_2	13707570	19798250	38206280	
4	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_3	5	5	5	
5	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_4	114492800	109111870	127051330	
6	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_5	0	0	0	
7	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_6	0,05	0,05	0,05	
8	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_7	0,000	0,000	0,000	



```
// benannten Zellbereich erzeugen
HSSFName namedCell = m_workbook.createName();
// Name des Tabellenblattes herausfinden
String sheetName = m_workbook.getSheetName(sheetIndex);
// Name des Zellbereichs setzen
namedCell.setNameName(wert);
// Referenz erstellen für die Zelle, so dass der Zellbereich auf die Zelle
// verweist
namedCell.setReference(sheetName+"!" + m_row.getRowNum() + m_cell.getCellNum());
```

- `getRowNum()` und `getCellNum()` liefern Zellposition in der Form „3,2“ zurück, also 3. Zeile, 2. Spalte, und nicht als „B3“
- `setReference()` benötigt den Input im Format „B3“

- Umwandlung des Formats

```
// aktuelle Zelle umwandeln von Format 0,0 in Excel-Format A1
String cellName = new CellReference(m_row.getRowNum(),
                                   m_cell.getCellNum()).toString();
```

- Umformulierung von setReference()

```
namedCell.setReference(sheetName+"!" + cellName);
```

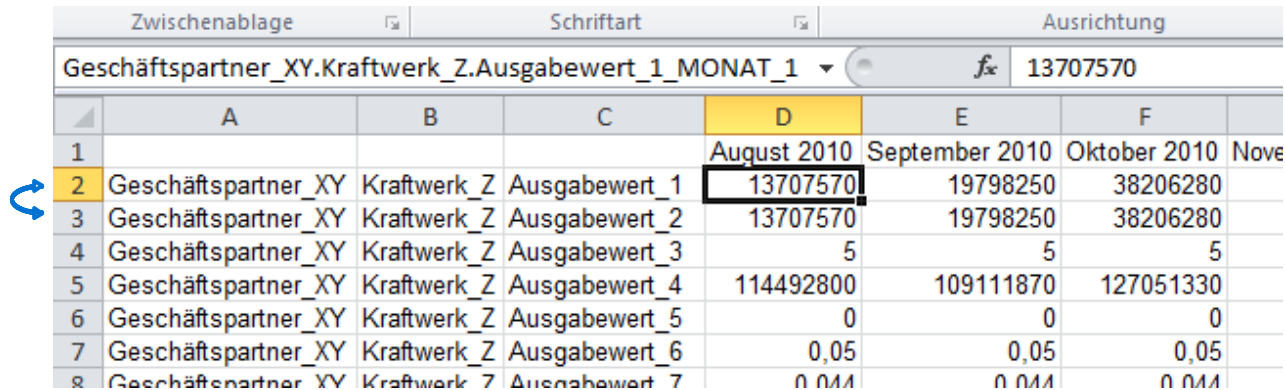
- „B3“ wird relativ interpretiert (3. Zeile, 2. Spalte vom aktuellen Ort)
  - es musste „\$B\$3“ übergeben werden
  - Zellname über regulären Ausdruck aufspalten

```
// benannten Zellbereich erzeugen
HSSFName namedCell = m_workbook.createName();
// Name des Tabellenblattes herausfinden
String sheetName = m_workbook.getSheetName(sheetIndex);
// Name des Zellbereichs setzen
namedCell.setNameName(wert);
// aktuelle Zelle umwandeln von Format 0,0 in Excel-Format A1
String cellName = new CellReference(m_row.getRowNum(),
                                   m_cell.getCellNum()).toString();
// Buchstabe heraussuchen, um $-Zeichen einfügen zu können
Pattern patt = Pattern.compile("[0-9]");
Matcher m = patt.matcher(cellName);
String cellLetter = m.replaceAll("");
int cellNumber = m_row.getRowNum()+1;
// Referenz erstellen für die Zelle, so dass der Zellbereich auf die Zelle
// verweist,
// $-Zeichen werden benötigt, da sonst die Region relativ und nicht absolut
// angelegt wird
namedCell.setReference(sheetName+"!$" + cellLetter + "$" + cellNumber);
```

- Bug: unmöglich, Zellbereich für eine einzelne Zelle zu definieren.
  - echter Bereich mit nur einer Zelle musste definiert werden

```
// Referenz erstellen für die Zelle, so dass der Zellbereich auf die Zelle
// verweist, wegen Bug nicht einzelne Zelle, sondern Bereich
// $-Zeichen werden benötigt, da sonst die Region relativ und nicht absolut
// angelegt wird
namedCell.setReference(sheetName+"!$" + cellLetter + "$" + cellNumber + ":" +
    cellLetter + "$" + cellNumber);
```

- Reihenfolge der Ausgabewerte einer Auswertung vertauscht



Zwischenablage		Schriftart		Ausrichtung			
Geschäftspartner_XY.Kraftwerk_Z.Ausgabewert_1_MONAT_1				fx	13707570		
	A	B	C	D	E	F	G
1				August 2010	September 2010	Oktober 2010	Nov
2	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_1	13707570	19798250	38206280	
3	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_2	13707570	19798250	38206280	
4	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_3	5	5	5	
5	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_4	114492800	109111870	127051330	
6	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_5	0	0	0	
7	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_6	0,05	0,05	0,05	
8	Geschäftspartner_XY	Kraftwerk_Z	Ausgabewert_7	0,001	0,001	0,001	

- „Ausgabewert 1“ → 3. Zeile, „Ausgabewert 2“ → 2. Zeile
- D2: „Geschäftspartner\_XY.Kraftwerk\_Z.Ausgabewert\_2\_MONAT\_1“
- D3: „Geschäftspartner\_XY.Kraftwerk\_Z.Ausgabewert\_1\_MONAT\_1“
- Namen über POI in der neuen Reihenfolge nach Excel
- Excel bekommt die geänderte Reihenfolge nicht mit

- Lösung:
    - im Vorfeld prüfen, in welcher Reihenfolge die Namen in der Excel-Vorlage angelegt worden waren (Reihenfolge über die ID)
    - beim Laden der Daten nach Excel Namen der Ergebnis-Zellen nicht zeitgleich mit dem Füllen der Daten anlegen,
    - Werte in einer anderen Reihenfolge als die zugehörigen Zellnamen nach Excel schreiben
- IDs in der richtigen Reihenfolge erhalten
- Verweise von der Excel-Vorlage korrekt in die eigentliche Auswertung übernommen.

- 2. festes Tabellenblatt: Ausgabe ganzer Zeitreihen
- Zeitreihen = Viertelstundenwerte  
→ je Zeitreihe und Monat bis zu 2976 Werte
- Auswertung
  - 20 Zeitreihen
  - 12 Monate
  - mehr als 700.000 Werte nach Excel

→ Dies brachte POI an seine Grenzen.

- Möglichkeit 1: mehr Speicher zuzuweisen
  - nicht möglich, da POI in der JVM von Forms mit läuft und somit darüber begrenzt ist.
- Möglichkeit 2: POI-Funktionalitäten aus der Form auszulagern
  - separater Thread
  - größeren Speicher mitgeben
- einfachste Möglichkeit: komplette Zeitreihen von so umfangreichen Auswertungen werden nicht benötigt (700.000 Werte!)
  - Zeitreihen aus den Auswertungen entfernt (Zuordnung in der Form)
  - Seiteneffekt: bessere Performance, weil wesentlich weniger Daten benötigt wurden.



- Ausgangslage
- fachlicher Hintergrund
- technischer Hintergrund
- POI-Framework
- Umsetzung
- Probleme und Lösungen
- Vergleich alt ↔ neu
- Ausblick

- wesentlicher Punkt bei der Neuentwicklung des Systems.
- Altsystem Client/Server-System
  - alle Auswertungen auf dem Client
  - viele Auswertungen sehr zeitintensiv
  - Funktionalität für Auswertungen auf dem Datenbank-Server
  - regelmäßige Prüfung nötig, ob die Auswertungen fertig waren
  - Abholung nach erfolgreicher Durchführung
- Umstellung auf Forms im Web
  - Auswertungen auf dem Application Server
  - Transfer auf den Client nach Erstellung
  - wesentlich schneller und komfortabler als vorher

- Programmierung der Auswertungs-Schnittstelle aufwändiger
  - Excel und Visual Basic im VORSYSTEM wesentlich integrierter nutzbar
- Schnittstelle allerdings nun sinnvoller gekapselt
  - Vorteil bei weiteren Technologiewechseln
  - leichtere Ergänzung von Funktionalitäten

- Ausgangslage
- fachlicher Hintergrund
- technischer Hintergrund
- POI-Framework
- Umsetzung
- Probleme und Lösungen
- Vergleich alt ↔ neu
- **Ausblick**

- Fallbeispiel: größere Oracle Forms/Reports-Anwendung
  - Möglichkeit, nur Teile der Anwendung, auf eine neue Technologie umzustellen,
  - Anwendung selbst kann erst mal unberührt bleiben
- anschließend Migration der Anwendung
- oder nur Reports im neuen Stil
  - wesentlich leichter zu warten

- Umformung der Reports in Excel-Auswertungen.
  - Definition von Object Types in der Datenbank, welche die Struktur der Reports abbilden.
  - Füllen der Object Types in der Datenbank mit Werten
  - Excel-Vorlage, vom Layout her dem bisherigen Report entsprechend
  - Vorlage mit den in der Datenbank zusammengestellten Werten verknüpfen
- Unterschied Excel – Reports:
  - Excel ist bearbeitbar durch den Anwender
  - kann über die Blattschutz-Funktionalität auch gesperrt werden
  - Option: teilweise bearbeitbar wie bei IAIS

- Technologiewechsel zu Java oder ADF nun wesentlich leichter
  - Schnittstelle gekapselt als Java-Funktionalität
  - ohne Probleme aus anderen Technologien aufrufbar



Vielen Dank!

?!

MT AG managing technology | Balcke-Dürr-Allee 9 | 40882 Ratingen  
Tel. +49 (0) 2102 309 61-0 | info@mt-ag.com | www.mt-ag.com







Di. 15.11.2011, 10 Uhr	Tabellen und Indizes reorganisieren, aber wann?	Ernst Leber
Di. 15.11.2011, 12 Uhr	Das Apache POI-Framework als Reporting-Tool für Oracle Anwendungen.	Sven-Olaf Kelbert
Di. 15.11.2011, 15 Uhr	AJAX-Rezepte für Web Services mit APEX-Bordmitteln, jQuery und JSON	Andreas Wismann
Mi. 16.11.2011, 9 Uhr	Gewusst wie: iPhone-Anbindung in SOA-Landschaften	Guido Neander
Mi. 16.11.2011, 10 Uhr	Tipps und Tricks in der Datenbankadministration	Volker Mach
Mi. 16.11.2011, 10 Uhr	Hybride mobile Applikationen mit ADF und PhoneGap	Christof Kaller
Mi. 16.11.2011, 14 Uhr	SAP+Microsoft+Social Media=Oracle WebCenter Portal.	Niels de Bruijn
Mi. 16.11.2011, 16 Uhr	Viel aus wenig: Enterprise-DWH mit Basic ETL	Jörg Menker
Do. 17.11.2011, 9 Uhr	Sesam (APEX) Öffne Deine Augen	Oliver Lemm
Do. 17.11.2011, 13 Uhr	Versionierung von SOA-Projekten mit Subversion im JDeveloper.	Klaus Friemelt
Do. 17.11.2011, 14 Uhr	Flexible Schriftstellen für Data Warehousing auf XML Basis.	Lutz Bauer
Do. 17.11.2011, 16 Uhr	DWH Migration nach Exadata: Performance „Out Of The Box“?	Lutz Bauer

Stand 330 (Tokio)