



# Oracle Forms, Microsoft and more

Gerd Volberg

OPITZ CONSULTING Deutschland GmbH

**Microsoft®**

**CONFLUENCE**

**JIRA**



Nürnberg, 20. November 2013

26	27	28
	Urlaub	

DOAG 2013

Showcase "E-Mail versenden"	<input type="button" value="Sende EMail"/>
Showcase "vCalendar erzeugen"	<input type="button" value="vCalendar"/> <input type="button" value="W"/>
Showcase "Jira-Ticket erzeugen"	<input type="button" value="Jira-Ticket"/>
Telefon Nr	<input type="text" value="02261 6001 0"/> <input type="button" value="Call"/>

# Agenda

---

- 1. Forms in die eigene Applikationslandschaft integrieren**
- 2. EMails erzeugen und verschicken**
- 3. Kalendereinträge für alle gängigen Kalender erzeugen**
- 4. Wiki-Systeme integrieren**
- 5. Issue-Tracking-Systeme einbinden**
- 6. Telefonanrufe initiieren**
- 7. ... and more**



# Forms in die eigene Applikationslandschaft integrieren



# **EMails erzeugen und verschicken**

# Basis-Funktionen

---

**Viele 3rd-Party Tools bieten Schnittstellen, die via EMail initialisiert und gesteuert werden können.**

**Die wichtigste Voraussetzung, solche Systeme anzusteuern, ist die Bereitstellung und Erzeugung von EMail aus der Oracle Datenbank heraus.**

**z.B.**

```
Sende_Email (P_VON           => 'formsapp@oc.de'  
             ,P_VON_NAME      => 'Forms-Applikation'  
             ,P_AN            => 'gerd.volberg@oc.de'  
             ,P_AN_NAME       => 'Gerd Volberg'  
             ,P_BETREFF       => 'Test-EMail'  
             ,P_NACHRICHT     => 'Hello World');
```

# Sourcecode Sende\_EMail (1)

---

```
PROCEDURE Sende_EMail
(P_VON          IN VARCHAR2
,P_VON_NAME     IN VARCHAR2 := NULL
,P_AN          IN VARCHAR2
,P_AN_NAME     IN VARCHAR2 := NULL
,P_CC          IN VARCHAR2 := NULL
,P_CC_NAME     IN VARCHAR2 := NULL
,P_BCC         IN VARCHAR2 := NULL
,P_BETREFF     IN VARCHAR2 := '(Betreff ist leer)'
,P_NACHRICHT   IN VARCHAR2 := '(Nachricht ist leer)'
,P_EMAIL_TYP   IN VARCHAR2 :=
    'Content-Type: text/plain; charset="ISO-8859-1"'
) IS
V_Host          VARCHAR2 (100);
V_Connection    utl_smtp.connection;
```

# Sourcecode Sende\_EMail (2)

```
PROCEDURE Schreibe_Daten (P_To_From_Str IN VARCHAR2,  
                        P_Name IN VARCHAR2, P_Email IN VARCHAR2) IS  
BEGIN  
    IF P_Name IS NOT NULL  
    AND P_Email IS NOT NULL THEN  
        utl_smtp.write_data (V_Connection, P_To_From_Str || ': ' ||  
                            P_Name || ' <' || P_Email || '>' || CHR (10));  
    ELSIF P_Email IS NOT NULL THEN  
        utl_smtp.write_data (V_Connection, P_To_From_Str || ': ' ||  
                            P_Email || CHR (10));  
    END IF;  
END;  
PROCEDURE Schreibe_Adressat (P_Email IN VARCHAR2) IS  
BEGIN  
    IF P_Email IS NOT NULL THEN  
        utl_smtp.rcpt (V_Connection, P_Email);  
    END IF;  
END;
```



# Sourcecode Sende\_Email (3)

---

```
BEGIN
  V_Host := 'mailrelay.oc.de';
  V_Connection := utl_smtp.open_connection (V_Host, 25);
  utl_smtp.helo (V_Connection, V_Host);
  utl_smtp.mail (V_Connection, P_Von);
  Schreibe_Adressat (P_An);
  Schreibe_Adressat (P_CC);
  Schreibe_Adressat (P_BCC);

  utl_smtp.open_data (V_Connection);

  Schreibe_Daten ('From', P_Von_Name, P_Von);
  Schreibe_Daten ('To', P_An_Name, P_An);
  Schreibe_Daten ('CC', P_CC_Name, P_CC);

  utl_smtp.write_data (V_Connection, 'Subject: ' || P_Betreff || CHR (10));
  utl_smtp.write_data (V_Connection, P_Email_Typ || CHR (10) || CHR (10));
  utl_smtp.write_data (V_Connection, P_Nachricht);
  utl_smtp.close_data (V_Connection);
  utl_smtp.quit (V_Connection);
END;
```



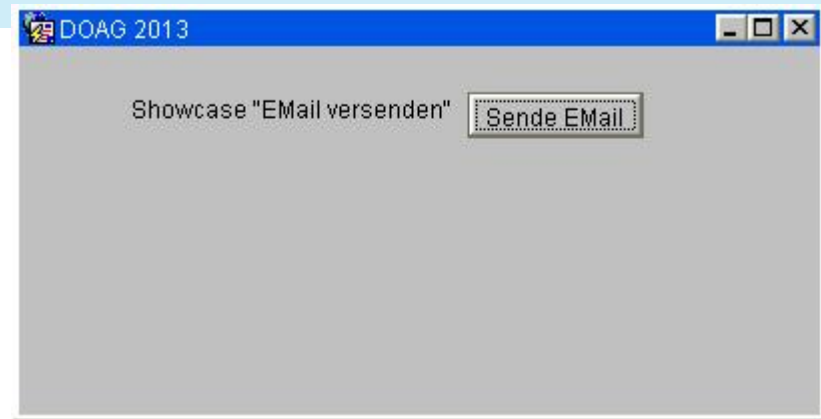


# Showcase EMail-Versand

## 1. Leeres Postfach



## 2. Klick auf Sende\_EMail

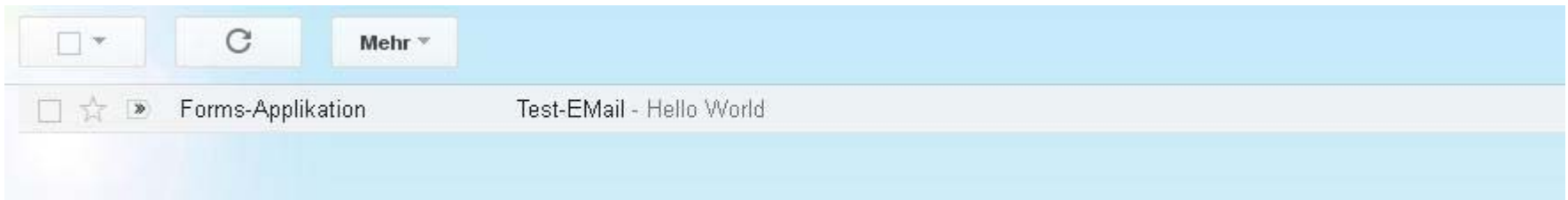


## 3. Sourcecode

```
Sende_EMail (P_VON           => 'formsapp@oc.de'  
             ,P_VON_NAME     => 'Forms-Applikation'  
             ,P_AN           => 'gerd.volberg@oc.de'  
             ,P_AN_NAME     => 'Gerd Volberg'  
             ,P_BETREFF     => 'Test-EMail'  
             ,P_NACHRICHT   => 'Hello World');
```



# Showcase EMail-Versand

## 4. Neue EMail



## 5. Inhalt

Test-EMail  DOAG 

 **Forms-Applikation** <formsapp@oc.de>  
an mich   
Hello World

 Zum [Antworten](#), [Allen antworten](#) oder [Weiterleiten](#) hier klicken



# Kalendereinträge für alle gängigen Kalender erzeugen

# Definition vCalendar

---

**Kalendereinträge, die per vCalendar erzeugt werden, sind heutzutage in allen gängigen Kalendern Standard.**

**EMails mit einer attachten vCalendar-Datei können vom User sofort in Kalendereinträge überführt werden.**



**WIKIPEDIA**  
Die freie Enzyklopädie

## vCalendar

---

**vCalendar** ist ein [Industriestandard](#) zum Austausch von Kalenderdaten und Termininformationen. Dieses Datenformat wird von [Software](#) auf [PCs](#), [PDAs](#) und [Mobiltelefonen](#) verwendet. *vCalendar* wurde von [iCalendar](#) abgelöst. Die Dateiendung von *vCalendar*-Dateien lautet „.vcs“.

*vCalendar* [spezifiziert](#) lediglich das [Format](#) der Daten, ohne eine bestimmte Methode der [Datenübermittlung](#) vorauszusetzen. Der Datenaustausch kann beispielsweise über [Dateien](#), per E-Mail oder drahtlos über eine [Infrarot-Schnittstelle](#) oder [Bluetooth](#) erfolgen.

# Basis-Funktionen

---

**Ein Kalendereintrag sollte im einfachsten Fall mit drei bis vier Parametern erzeugt werden können:**

```
Sende_vCalendar (  
  P_DATUM_START      => to_date ('27.11.2013', 'DD.MM.YYYY')  
,P_DATUM_ENDE      => to_date ('27.11.2013', 'DD.MM.YYYY')  
,P_AN              => 'gerd.volberg@oc.de'  
,P_BETREFF         => 'Urlaub');
```

# Sourcecode Sende\_vCalendar (1)

---

```
PROCEDURE Sende_vCalendar
(P_DATUM_START      IN DATE
,P_DATUM_ENDE      IN DATE          := NULL
,P_AN              IN VARCHAR2
,P_BETREFF         IN VARCHAR2
,P_ORT             IN VARCHAR2      := NULL
,P_BESCHREIBUNG    IN VARCHAR2      := NULL
,P_GANZTAEGIG_KNZ  IN VARCHAR2      := 'J'
) IS

V_DATUM_ENDE      DATE;
V_CLOB            CLOB;
```

# Sourcecode Sende\_vCalendar (2)

```
FUNCTION Format_Date (P_Datum IN DATE) RETURN VARCHAR2 IS
  V_DELTA_STUNDEN NUMBER(2) := 2; -- Timezone-Korrektur
BEGIN
  IF P_GANZTAEGIG_KNZ = 'J' THEN
    RETURN (';VALUE=DATE:' || TO_CHAR (P_Datum, 'YYYYMMDD'));
  ELSE
    RETURN (':' || TO_CHAR (P_Datum-V_Delta_Stunden/24, 'YYYYMMDD') ||
            'T' || TO_CHAR (P_Datum-V_Delta_Stunden/24, 'HH24MISS') ||
            'Z');
  END IF;
END;
BEGIN
  IF P_Datum_Endes IS NULL THEN
    V_Datum_Endes := P_Datum_Start;

  ELSIF P_Datum_Start = TRUNC (P_Datum_Start)
  AND P_Datum_Endes = TRUNC (P_Datum_Endes)
  AND P_Datum_Endes > P_Datum_Start THEN
    V_Datum_Endes := P_Datum_Endes + 1;
  END IF;
```

Ein Zeitraum wird jeweils von 0 Uhr bis 0 Uhr gespeichert. Deswegen muss das BIS-Datum um 1 Tag vergrössert werden.

# Sourcecode Sende\_vCalendar (3)

```
V_CLOB := 'BEGIN:VCALENDAR' || CHR (10) ||
         'VERSION:2.0' || CHR (10) ||
         'BEGIN:VEVENT' || CHR (10) ||
         'DESCRIPTION:' || P_Beschreibung || CHR (10) ||
         'DTEND' || Format_Date (V_Datum_Ende) || CHR (10) ||
         'DTSTART' || Format_Date (P_Datum_Start) || CHR (10) ||
         'LOCATION:' || P_Ort || CHR (10) ||
         'SUMMARY:' || P_Betreff || CHR (10) ||
         'TRANSP:OPAQUE' || CHR (10) ||
         'X-MICROSOFT-CDO-BUSYSTATUS:OOF' || CHR (10) ||
         'END:VEVENT' || CHR (10) ||
         'END:VCALENDAR';

frw_Email.Send_CLOB (
    P_VON      => 'formsapp@oc.de',
    P_AN       => P_An,
    P_Betreff  => P_Betreff,
    P_Dateiname => 'Kalendereintrag.vcs',
    P_Nachricht => V_CLOB);

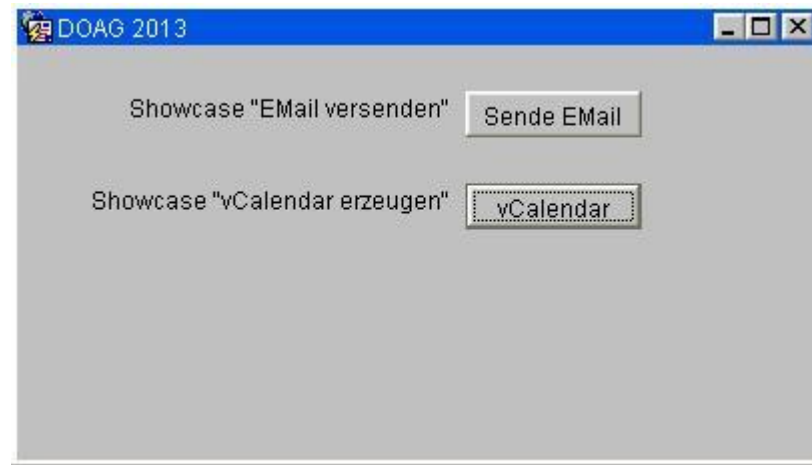
END;
```





# Showcase vCalendar

## 1. Klick auf vCalendar



## 2. Outlook





# Wiki-Systeme integrieren

# Wiki-Systeme

---



DokuWiki  
It's better when it's simple



# Vorbereitungen

---

- 1. Im folgenden Beispiel sind alle Wikiseiten unter folgender URL erreichbar:**

`https://confluence.oc.de/display/Formsapp/<Name-der-Wiki-Seite>`

- 2. Der statische Teil der URL sollte an zentraler Stelle definiert werden, so dass ein Zugriff möglichst einfach gewährleistet werden kann.**

Z.B. in einer Parametertabelle oder einer Konstanten

```
C_Startseite_Wiki := 'https://confluence.oc.de/display/Formsapp/';
```

# Vorbereitungen

**Zwei Hilfstabellen speichern die Metadaten der Formsmasken und der korrespondierenden Wikiseiten**

## Tabelle FORMS

FORM_ID	1	1 N	NUMBER (9)	PK
FORM_FILE_NAME	2	N	VARCHAR2 (14 Byte)	Phys. Name der Datei
FORM_NAME	3	N	VARCHAR2 (30 Byte)	Name der Maske

## Tabelle WIKIS

WIKI_ID	1	1 N	NUMBER (9)	PK
WIKI_FORM_ID	2	N	NUMBER (9)	FK auf Forms
WIKI_ITEM_NAME	3	Y	VARCHAR2 (100 Byte)	Itemname
WIKI_OFFSET	4	Y	VARCHAR2 (100 Byte)	URL-Offset, bezogen auf die Start-Wiki-Seite

# Sourcecode Starte\_Wiki

---

```
PROCEDURE Starte_Wiki IS
    V_FORM_Name          VARCHAR2 (30);
    V_ITEM_Name          VARCHAR2 (61);
    V_Offset              WIKIS.WIKI_Offset%TYPE;
    C_Startseite_Wiki    CONSTANT VARCHAR2 (100)
        := 'https://confluence.oc.de/display/Formsapp/';
BEGIN
    V_FORM_Name := NAME_IN ('System.Current_Form');
    V_ITEM_Name := NAME_IN ('System.Trigger_Item');

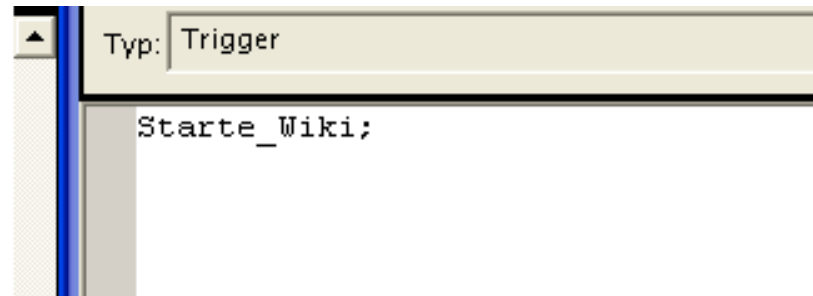
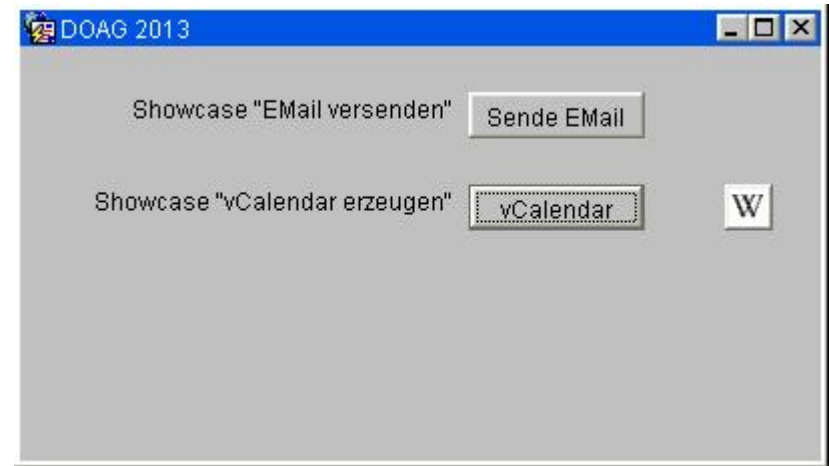
    SELECT WIKI_Offset
        INTO V_Offset
        FROM Wikis, Forms
        WHERE WIKI_FORM_ID = FORM_ID
            AND FORM_NAME = V_FORM_Name
            AND WIKI_Item_Name = V_ITEM_Name;

    WEB.Show_Document (C_Startseite_Wiki || V_Offset);
END;
```



# Showcase Wiki

## Klick auf den Wiki-Button



# Showcase Wiki

## Ergebnis



The screenshot shows a web browser window displaying a Confluence page. The address bar shows the URL: <https://confluence.oc.de/display/Formsapp/DOAG+2013>. The Confluence navigation bar includes the logo, the text 'Confluence', and links for 'Bereiche', 'Personen', and 'Erstellen'. The page content includes a breadcrumb 'Seiten / Hauptseite', the title 'DOAG 2013', and a note: '1 Hinzugefügt von Volberg, Gerd, zuletzt bearbeitet von Volberg, Gerd am 12.11.2013 (Änderung anzeigen)'. Below this, the text reads: 'Diese Maske dient als Showcase für diverse Demos auf der DOAG 2013'. An embedded window titled 'DOAG 2013' displays two showcase items: 'Showcase "E-Mail versenden"' with a 'Sende EMail' button, and 'Showcase "vCalendar erzeugen"' with a 'vCalendar' button and a 'W' button.





# Issue-Tracking-Systeme einbinden

# Definition Issue-Tracking

---

**Moderne Forms-Applikationen sollten immer eine Schnittstelle zum firmeneigenen Issue-Tracking besitzen.**

**Im nachfolgenden Beispiel wird dies anhand von Jira demonstriert**



## Issue-Tracking-System

---

Ein **Issue-Tracking-System** (ITS; Synonyme: [Helpdesk-System](#), [Serviceticket-System](#), [Ticketing-System](#), [Task-Tracking-System](#), [Support-Ticketing-System](#), [Trouble-Ticket-System](#), [Request-Tracking-System](#) (RTS), teilweise auch [Fallbearbeitungssystem](#)) ist eine Art von Software, um Empfang, Bestätigung, Klassifizierung und Bearbeitung von Kundenanfragen (Tickets bzw. Fälle) zu handhaben. Als Anfragen werden eingehende Kundenanrufe, E-Mails, Faxe und Ähnliches betrachtet.

# Sourcecode Create\_Jira\_Ticket (1)

---

```
PROCEDURE Create_Jira_Ticket
(P_VON                IN VARCHAR2
,P_VON_NAME          IN VARCHAR2 := NULL
,P_BETREFF           IN VARCHAR2
,P_PROJEKT            IN VARCHAR2
,P_VORGANGSART        IN VARCHAR2 := 'Anfrage'
,P_PRIO               IN VARCHAR2 := 'Normal'
,P_KOMPONENTE         IN VARCHAR2 := NULL
,P_SICHERHEITSSTUFE  IN VARCHAR2 := NULL
,P_TEXT              IN VARCHAR2 ) IS

V_Komponente          VARCHAR2(100);
V_Sicherheitsstufe    VARCHAR2(100);
```

# Sourcecode Create\_Jira\_Ticket (2)

```
BEGIN
IF P_Komponente IS NOT NULL THEN
  V_Komponente := '@Komponente = ' || P_Komponente || CHR (10);
END IF;
IF P_Sicherheitsstufe IS NOT NULL THEN
  V_Sicherheitsstufe := '@Sicherheitsstufe = ' ||
    P_Sicherheitsstufe || CHR (10);
END IF;
Sende_Email (P_Von => P_Von,          P_Von_Name => P_Von_Name,
             P_An  => 'jira@oc.de', P_An_Name  => 'Jira',
             P_Betreff => P_Betreff,
             P_Nachricht =>
               '@Projekt = '          || P_Projekt          || CHR (10) ||
               '@Vorgangsart = '     || P_Vorgangsart     || CHR (10) ||
               '@Prio = '             || P_Prio             || CHR (10) ||
               V_Komponente          || V_Sicherheitsstufe ||
               P_Text);
END;
```



# Showcase Issue-Tracking

## Manuell erzeugtes Ticket



### Vorgang erstellen

Projekt*	<input type="text" value="YAPUT"/>	▼
Vorgangstyp*	<input type="text" value="Fehler"/>	▼ ⓘ
Priorität	<input type="text" value="Normal"/>	▼ ⓘ
Sicherheitsstufe	<input type="text" value="Keine"/>	▼ ⓘ
Komponente(n)	<b>Keine</b>	
Zusammenfassung*	<input type="text" value="Issue-Ticket-Test für die DOAG Konferenz"/>	
Beschreibung	<input type="text" value="Dieses Ticket wurde in Jira manuell erzeugt."/>	

# Showcase Issue-Tracking

## Automatisch erzeugtes Ticket



```
Create_Jira_Ticket (  
    P_Von           => 'gerd.volberg@oc.de',  
    P_Betreff       => 'Automatisch erzeugtes Ticket',  
    P_Projekt       => 'YAPUT',  
    P_Text          => 'Aus Forms erzeugtes Ticket.');
```

## Ergebnis

Schlüssel Zusammenfassung

Bearbeiter

YAPUT-3 Automatisch erzeugtes Ticket

*Nicht zugewiesen*

YAPUT-2 Issue-Ticket-Test für die DOAG Konferenz

*Nicht zugewiesen*



# Telefonanrufe initiieren

# Integration einer Telefonanlage

**Aktuelle Telefonanlagen lassen sich häufig per API fernbedienen. Im folgenden Beispiel wird dies anhand der Telefonanlage von Siemens Open Scape demonstriert**

**Im Standardumfang wird ein Tomcat mitgeliefert, der per Webschnittstelle automatisiert Anrufe initiieren kann**





# Vorbereitungen

---

- 1. Im folgenden Beispiel werden Anrufe unter folgender URL initiiert**

`https://osucapp.uc.oc.de/tweb/portal?clickToDial=<Telefonnummer>`

- 2. Der statische Teil der URL sollte wieder an zentraler Stelle definiert werden:**

Z.B. in einer Parametertabelle oder einer Konstanten

```
C_Telefon_URL := 'https://osucapp.uc.oc.de/tweb/portal?clickToDial=';
```

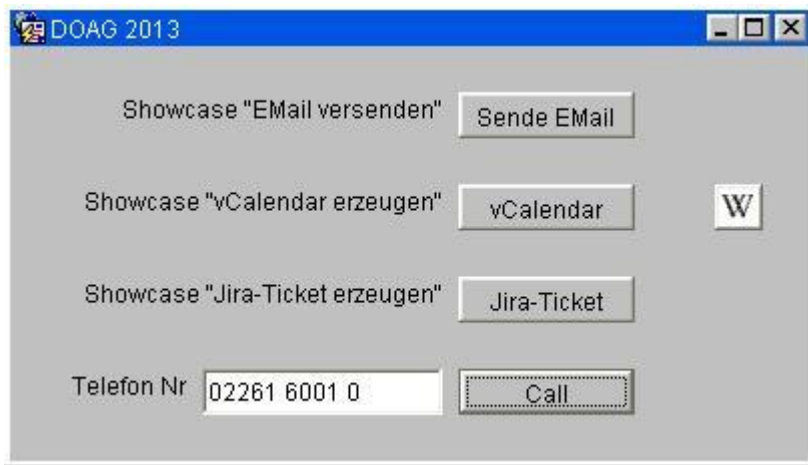
# Sourcecode Telefon\_Direktwahl

---



```
PROCEDURE Telefon_Direktwahl (P_Telefon_String IN VARCHAR2) IS
BEGIN
    WEB.Show_document (C_Telefon_URL ||
                      REPLACE (P_Telefon_String, ' ', NULL));
END;
```

# Showcase Telefonanruf initiieren



DOAG 2013

Showcase "EMail versenden"

Showcase "vCalendar erzeugen"

Showcase "Jira-Ticket erzeugen"

Telefon Nr

```
Telefon_Direktwahl  
(:Control.TI_Telefon_Nummer);
```



OpenScope WebClient Journal - Mozilla Firefox

https://osucapp.uc... /tweb/portal/req;jsessionid=4F26783B9B9953CA93E6

Journal

Pro Anrufer  Alle Anrufe

Datum	Dauer	Name	Nummer
18.11 15:42	0:00	?	0226160011202

Einträge: 1 - 1 (...)



**... and more**

# CSV-Dateien erzeugen und verschicken

**Wir benötigen dazu eine Hilfsfunktion, die EMail mit Attachments versendet.**

**Diese Attachments sind in diesem Fall nun CSV-Dateien**

**Eine weitere Hilfsfunktion wird einen CLOB aufbauen, der die Rohdaten aufnimmt, die dann per EMail versendet werden.**

W.	Datum	von	bis	Pause	Fakt.	Min.	Kunde	Kat.
mo	04.11.2013	08:00	17:00	30	N	510	OPITZ CONSULTING	RN1
di	05.11.2013	06:00	15:00	30	N	510	OPITZ CONSULTING	INT
mi	06.11.2013	07:15	16:00	30	N	495	OPITZ CONSULTING	INT
do	07.11.2013	08:45	17:30	30	N	495	OPITZ CONSULTING	INT
fr	08.11.2013	06:45	12:30	30	N	315	OPITZ CONSULTING	RN1
fr	08.11.2013	12:30	16:00	45	N	165	OPITZ CONSULTING	INT
mo	11.11.2013	07:00	16:30	30	N	540	OPITZ CONSULTING	RN1
di	12.11.2013	06:00	16:00	30	N	570	OPITZ CONSULTING	RN1
mi	13.11.2013	06:15	16:00	30	N	555	OPITZ CONSULTING	RN1
do	14.11.2013	07:15	16:00	30	N	495	OPITZ CONSULTING	INT
fr	15.11.2013	08:00	16:00	0	N	480	OPITZ CONSULTING	INT
mo	18.11.2013	08:30	23:00	30	N	840	OPITZ CONSULTING	RN1

# Sourcecode frw\_EMail

---

Konstante CSV\_EXPORT\_ZEICHENSATZ wird benötigt !

```
CREATE OR REPLACE PACKAGE BODY frw_EMail IS
```

```
PROCEDURE CLOSE
```

```
(P_CONNECTION IN OUT utl_smtp.connection ) IS
```

```
BEGIN
```

```
    utl_smtp.write_data (P_Connection, utl_tcp.crlf);
```

```
    utl_smtp.close_data (P_Connection);
```

```
    utl_smtp.quit      (P_Connection);
```

```
END;
```

# Sourcecode frw\_EMail

---

```
PROCEDURE INITIALIZE
(P_VON IN VARCHAR2 ,P_AN IN VARCHAR2
,P_BETREFF IN VARCHAR2 ,P_CONNECTION OUT utl_smtp.connection ) IS
V_HOST VARCHAR2(100);
BEGIN
  IF P_Von IS NOT NULL
  AND P_An IS NOT NULL THEN
    V_Host      := 'mailrelay.oc.de';
    P_Connection := utl_smtp.open_connection (V_Host, 25);

    utl_smtp.helo (P_Connection, V_Host);
    utl_smtp.mail (P_Connection, P_Von);
    utl_smtp.rcpt (P_Connection, P_An);

    utl_smtp.open_data      (P_Connection);
    utl_smtp.write_data     (P_Connection, 'From: ' || P_Von || utl_tcp.crlf);
    utl_smtp.write_data     (P_Connection, 'To: '   || P_An  || utl_tcp.crlf);
    utl_smtp.write_raw_data (P_Connection, utl_raw.cast_to_raw
                                ('Subject: ' || P_Betreff || utl_tcp.crlf));

  END IF;
END;
```



# Sourcecode frw\_EMail

---

```
PROCEDURE INITIALIZE_WITH_ATTACHMENT
(P_VON IN VARCHAR2 ,P_AN IN VARCHAR2
,P_BETREFF IN VARCHAR2 ,P_DATEINAME IN VARCHAR2
,P_CONNECTION OUT utl_smtp.connection ) IS
V_CONNECTION UTL_SMTP.CONNECTION;
BEGIN
  Initialize (P_Von          => P_Von,
             P_An           => P_An,
             P_Betreff      => P_Betreff,
             P_Connection => V_Connection);

  utl_smtp.write_raw_data (V_Connection, utl_raw.cast_to_raw
                          ('Content-Disposition: attachment; filename="' ||
                           P_Dateiname || '"' || utl_tcp.crlf));
  utl_smtp.write_data    (V_Connection, 'Content-Transfer-Encoding: base64' ||
                          utl_tcp.crlf);
  utl_smtp.write_data    (V_Connection, utl_tcp.crlf);

  P_Connection := V_Connection;
END;
```





# Sourcecode frw\_EMail

---

```
PROCEDURE SEND
(P_VON IN VARCHAR2 ,P_AN IN VARCHAR2
,P_BETREFF IN VARCHAR2 ,P_NACHRICHT IN VARCHAR2 := NULL ) IS
V_CONNECTION UTL_SMTP.CONNECTION;
BEGIN
  Initialize (P_Von      => P_Von,
             P_An       => P_An,
             P_Betreff  => P_Betreff,
             P_Connection => V_Connection);

  utl_smtp.write_data      (V_Connection, 'Content-Type: text/html;charset=utf-8 '
                          || utl_tcp.crlf);
  utl_smtp.write_raw_data (V_Connection, utl_raw.cast_to_raw (utl_tcp.crlf
                          || P_Nachricht || utl_tcp.crlf));

  utl_smtp.close_data     (V_Connection);
  utl_smtp.quit           (V_Connection);

EXCEPTION
  WHEN utl_smtp.transient_error
  OR   utl_smtp.permanent_error THEN
    utl_smtp.quit (V_Connection);
    RAISE;

  WHEN OTHERS THEN
    RAISE;

END;
```

# Sourcecode frw\_EMail

---

```
PROCEDURE SEND_BLOB_FROM_DB
  (P_VON IN VARCHAR2 ,P_AN IN VARCHAR2
  ,P_BETREFF IN VARCHAR2 ,P_DATEINAME IN VARCHAR2
  ,P_BLBS_ID IN NUMBER ) IS
V_CONNECTION UTL_SMTP.CONNECTION;
V_BLBS BLOBS%ROWTYPE;
BEGIN
  V_BLBS := BLBS.Get_Row (P_BLBS_ID);
  Send_BLOB (P_Von      => P_Von,
            P_An       => P_An,
            P_Betreff  => P_Betreff,
            P_Dateiname => P_Dateiname,
            P_BLOB     => V_BLBS.BLBS_BLOB);

EXCEPTION
  WHEN OTHERS THEN
    RAISE;

END;
```



# Sourcecode frw\_EMail

---

```
PROCEDURE SEND_BLOB
(P_VON IN VARCHAR2 ,P_AN IN VARCHAR2
,P_BETREFF IN VARCHAR2 ,P_DATEINAME IN VARCHAR2
,P_BLOB IN BLOB ) IS
V_CONNECTION UTL_SMTP.CONNECTION;
BEGIN
    Initialize (P_Von          => P_Von,
               P_An           => P_An,
               P_Betreff      => P_Betreff,
               P_Connection => V_Connection);

    utl_smtp.write_raw_data (V_Connection, utl_raw.cast_to_raw
                              ('Content-Disposition: attachment;
                               filename="' || P_Dateiname || '"' || utl_tcp.crlf));
    utl_smtp.write_data      (V_Connection, 'Content-Transfer-Encoding: base64' ||
                              utl_tcp.crlf);

    utl_smtp.write_data      (V_Connection, utl_tcp.crlf);
    utl_smtp.write_raw_data  (V_Connection, utl_encode.base64_encode (P_BLOB));
    utl_smtp.write_data      (V_Connection, utl_tcp.crlf);
    utl_smtp.write_data      (V_Connection, utl_tcp.crlf);
    utl_smtp.close_data      (V_Connection);
    utl_smtp.quit             (V_Connection);
END;
```



# Sourcecode frw\_EMail

---

```
PROCEDURE SEND_CLOB
(P_VON IN VARCHAR2 ,P_AN IN VARCHAR2
,P_BETREFF IN VARCHAR2 ,P_DATEINAME IN VARCHAR2
,P_CLOB IN CLOB ) IS
V_CONNECTION UTL_SMTP.CONNECTION;
BEGIN
  Initialize (P_Von          => P_Von,
              P_An           => P_An,
              P_Betreff      => P_Betreff,
              P_Connection => V_Connection);

  utl_smtp.write_raw_data (V_Connection, utl_raw.cast_to_raw ('Content-Disposition:
                          attachment; filename="' || P_Dateiname || '"' || utl_tcp.crlf));
  utl_smtp.write_data      (V_Connection, 'Content-Transfer-Encoding: base64' ||
                          utl_tcp.crlf);

  utl_smtp.write_data      (V_Connection, utl_tcp.crlf);
  utl_smtp.write_data      (V_Connection, utl_encode.text_encode (P_CLOB,
                          CSV_EXPORT_ZEICHENSATZ, utl_encode.base64));

  utl_smtp.write_data      (V_Connection, utl_tcp.crlf);
  utl_smtp.write_data      (V_Connection, utl_tcp.crlf);
  utl_smtp.close_data      (V_Connection);
  utl_smtp.quit            (V_Connection);
END;
```



# Sourcecode frw\_Email

---

```
PROCEDURE WRITE_CLOB
```

```
(P_CONNECTION IN OUT utl_smtp.connection  
,P_CLOB IN CLOB ) IS
```

```
BEGIN
```

```
    utl_smtp.write_data (P_Connection, utl_encode.text_encode  
(P_CLOB, CSV_EXPORT_ZEICHENSATZ , utl_encode.base64));  
END;
```

```
PROCEDURE WRITE_CLOB_CRLF
```

```
(P_CONNECTION IN OUT utl_smtp.connection  
,P_CLOB IN CLOB ) IS
```

```
BEGIN
```

```
    frw_Email.Write_CLOB (P_Connection => P_Connection,  
                          P_CLOB      => P_CLOB);
```

```
    utl_smtp.write_data (P_Connection, utl_tcp.crlf);
```

```
END;
```

# Sourcecode frw\_EMail

---

```
PROCEDURE WRITE_RAW
```

```
(P_CONNECTION IN OUT utl_smtp.connection  
,P_STRING IN VARCHAR2 ) IS
```

```
BEGIN
```

```
    utl_smtp.write_data (P_Connection, utl_encode.text_encode  
    (P_String, CSV_EXPORT_ZEICHENSATZ, utl_encode.base64));
```

```
END;
```

```
PROCEDURE WRITE_RAW_CLOB
```

```
(P_CONNECTION IN OUT utl_smtp.connection  
,P_CLOB IN CLOB ) IS
```

```
BEGIN
```

```
    utl_smtp.write_data (P_Connection, utl_encode.text_encode  
    (P_CLOB, CSV_EXPORT_ZEICHENSATZ, utl_encode.base64));
```

```
END;
```

# Sourcecode frw\_EMail

---

```
PROCEDURE WRITE_STRING
(P_CONNECTION IN OUT utl_smtp.connection
,P_STRING IN VARCHAR2 ) IS
BEGIN
    utl_smtp.write_data (P_Connection, utl_encode.text_encode
(P_String, CSV_EXPORT_ZEICHENSATZ, utl_encode.base64));
END;
```

```
PROCEDURE WRITE_STRING_CRLF
(P_CONNECTION IN OUT utl_smtp.connection
,P_STRING IN VARCHAR2 ) IS
BEGIN
    frw_EMail.Write_String (P_Connection => P_Connection,
                            P_String      => P_String);
    utl_smtp.write_data (P_Connection, utl_tcp.crlf);
END;
END frw_EMail;
```

# 1 \* Durchatmen

---



# Sourcecode frw\_CSV

---

```
CREATE OR REPLACE PACKAGE BODY frw_CSV IS

V_STRING VARCHAR2(4000);
V_TRENNZEICHEN VARCHAR2(10);
V_CLOB CLOB;

FUNCTION GET_CLOB_FROM_ROW
(P_CSV_ROW IN PVS_TYPE.T_CSV_ROW ) RETURN CLOB IS
BEGIN
    V_String := NULL;

    Add_Column (P_CSV_Row.Spalte_01);
    Add_Column (P_CSV_Row.Spalte_02);
    . . .
    Add_Column (P_CSV_Row.Spalte_49);
    Add_Column (P_CSV_Row.Spalte_50);

    RETURN (V_String || SPU_Const.CR);
END;
```

# Sourcecode frw\_CSV

---

```
FUNCTION GET_CLOB_LENGTH RETURN NUMBER IS
BEGIN
    RETURN (NVL (LENGTH (V_CLOB), 0));
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN (NULL);
END;
```

```
PROCEDURE ADD_COLUMN
(P_STRING IN VARCHAR2 ) IS
BEGIN
    V_String := V_String ||
                REPLACE (P_String, CHR (10), ' ') ||
                frw_CSV.V_Trennzeichen;
END;
```



# Sourcecode frw\_CSV

---

```
PROCEDURE ADD_ROW
  (P_CSV_ROW IN PVS_TYPE.T_CSV_ROW ) IS
BEGIN
  V_String := NULL;

  Add_Column (P_CSV_Row.Spalte_01);
  Add_Column (P_CSV_Row.Spalte_02);
  . . .
  Add_Column (P_CSV_Row.Spalte_49);
  Add_Column (P_CSV_Row.Spalte_50);

  V_CLOB := V_CLOB || V_String || SPU_Const.CR;
END;
```

# Sourcecode frw\_CSV

---

```
PROCEDURE ADD_ROW
  (P_SPALTE_01 IN VARCHAR2
  ,P_SPALTE_02 IN VARCHAR2 := NULL
  . . .
  ,P_SPALTE_49 IN VARCHAR2 := NULL
  ,P_SPALTE_50 IN VARCHAR2 := NULL ) IS
BEGIN
  V_String := NULL;

  Add_Column (P_Spalte_01);
  Add_Column (P_Spalte_02);
  . . .
  Add_Column (P_Spalte_49);
  Add_Column (P_Spalte_50);

  V_CLOB := V_CLOB || V_String || SPU_Const.CR;
END;
```



# Sourcecode frw\_CSV

---

```
PROCEDURE DESTROY IS
BEGIN
    V_CLOB := NULL;
END;
```

```
PROCEDURE INITIALIZE
(P_TRENNZEICHEN IN VARCHAR2 := ';' ) IS
BEGIN
    V_CLOB          := NULL;
    V_TRENNZEICHEN := P_TRENNZEICHEN;
END;
```

# Sourcecode frw\_CSV

---

```
PROCEDURE SEND_CLOB
(P_VON IN VARCHAR2 ,P_AN IN VARCHAR2
,P_BETREFF IN VARCHAR2
,P_DATEINAME IN VARCHAR2
,P_NACHRICHT IN VARCHAR2 ) IS
V_CONNECTION UTL_SMTP.CONNECTION;
BEGIN
frw_Email.Initialize_with_Attachment (P_Von      => P_Von,
                                       P_An        => P_An,
                                       P_Betreff    => P_Betreff,
                                       P_Dateiname  => P_Dateiname,
                                       P_Connection => V_Connection);
frw_Email.Write_CLOB_CRLF (P_Connection => V_Connection,
                           P_CLOB       => frw_CSV.V_CLOB);
frw_Email.Close (V_Connection);
frw_CSV.Destroy;
END;
END frw_CSV;
```

# 2 \* Durchatmen

---

# Loop zum Erzeugen einer CSV-Datei

```
PROCEDURE Versende_CSV_als_Email IS
  V_Dateiname      VARCHAR2 (2000) := 'Aufwandsdaten.csv';
  V_CLOB           CLOB;
BEGIN
  Append_Row_to_CSV_CLOB (V_CLOB,      'Datum', 'von', 'bis', 'Pause',
                           'Fakt.', 'Min.', 'Kunde', 'Kat. ');
  FOR R IN (SELECT * FROM Aufwaende_V WHERE AUFW_Datum > trunc (sysdate, 'MM'))
  LOOP
    Append_Row_to_CSV_CLOB (V_CLOB, to_char (R.AUFW_Datum, 'DD.MM.YYYY'),
                           to_char (R.AUFW_von, 'HH24:MI'),
                           to_char (R.AUFW_bis, 'HH24:MI'),
                           to_char (R.AUFW_Pause),
                           R.AUFW_Fakturierbar_KNZ,
                           to_char (R.AUFW_Dauer_Arbeit),
                           R.ADRE_Firmenname,
                           R.KATE_Matchcode);

  END LOOP;
  frw_Email.Send_CLOB (P_VON      => 'formsapp@oc.de',
                      P_AN        => 'gerd.volberg@oc.de',
                      P_Betreff    => 'Aufwands-Daten aktueller Monat',
                      P_Dateiname  => V_Dateiname,
                      P_CLOB       => V_CLOB);
END;
```





# CSV-Rohdaten (in Excel geöffnet)

W.	Datum	von	bis	Pause	Fakt.	Min.	Kunde	Kat.
mo	04.11.2013	08:00	17:00	30	N	510	OPITZ CONSULTING	RN1
di	05.11.2013	06:00	15:00	30	N	510	OPITZ CONSULTING	INT
mi	06.11.2013	07:15	16:00	30	N	495	OPITZ CONSULTING	INT
do	07.11.2013	08:45	17:30	30	N	495	OPITZ CONSULTING	INT
fr	08.11.2013	06:45	12:30	30	N	315	OPITZ CONSULTING	RN1
fr	08.11.2013	12:30	16:00	45	N	165	OPITZ CONSULTING	INT
mo	11.11.2013	07:00	16:30	30	N	540	OPITZ CONSULTING	RN1
di	12.11.2013	06:00	16:00	30	N	570	OPITZ CONSULTING	RN1
mi	13.11.2013	06:15	16:00	30	N	555	OPITZ CONSULTING	RN1
do	14.11.2013	07:15	16:00	30	N	495	OPITZ CONSULTING	INT
fr	15.11.2013	08:00	16:00	0	N	480	OPITZ CONSULTING	INT
mo	18.11.2013	08:30	23:00	30	N	840	OPITZ CONSULTING	RN1

# Ihr Ansprechpartner

**Gerd Volberg**

Solution Architect

OPITZ CONSULTING Deutschland GmbH

Kirchstr. 6, 51647 Gummersbach

Tel. +49 (2261) 60 01-0

gerd.volberg@opitz-consulting.com



talk2gerd@blogspot.com



2. OKTOBER 2013

## Modernizing Forms Vortrag

Das Interview, über das ich vorgestern schrieb, ist in Zusammenarbeit mit meinem Vortrag "Wie modernisiere ich Oracle Forms" entstanden.

Das vollständige Video zu dem Vortrag (50 min) ist hier zu finden:

ÜBER MICH



GERD VOLBERG

Folgen 150

MEIN PROFIL