

# HERZLICH WILLKOMMEN

DOAG Development 2013  
Bonn

BI Publisher Integration  
über Oracle Datenbanken

Perry Pakull

19.06.2013

BASEL    BERN    LAUSANNE    ZÜRICH    DÜSSELDORF    FRANKFURT A.M.    FREIBURG I.BR.    HAMBURG    MÜNCHEN    STUTTGART    WIEN

1

2012 © Trivadis

BI Publisher Integration über Oracle Datenbanken  
19.06.2013

**trivadis**  
makes IT easier. ■ ■ ■

## About me



- Perry Pakull
  - Trivadis AG
  - Principal Consultant
- Oracle Application Development
  - SQL und PL/SQL
  - Forms und Reports
  - APEX
  - BI Publisher
- Architektur
  - System Design
  - Modernisierung



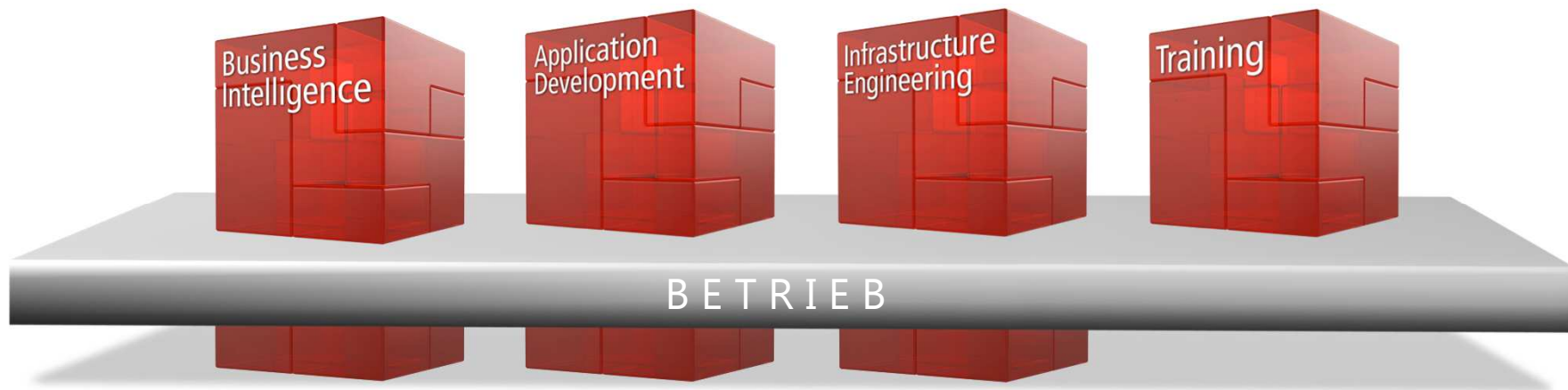
2012 © Trivadis

BI Publisher Integration über Oracle Datenbanken  
19.06.2013

# Unser Unternehmen

Trivadis ist **führend bei der IT-Beratung, der Systemintegration, dem Solution-Engineering** und der Erbringung von **IT-Services** mit Fokussierung auf **ORACLE®** und  **Microsoft** Technologien im D-A-CH-Raum.

Unsere Leistungen erbringen wir aus den strategischen Geschäftsfeldern:



Trivadis Services übernimmt den korrespondierenden Betrieb Ihrer IT Systeme.

## Mit über 600 IT- und Fachexperten bei Ihnen vor Ort



11 Trivadis Niederlassungen mit über 600 Mitarbeitenden

200 Service Level Agreements

Mehr als 4'000 Trainingsteilnehmer

Forschungs- und Entwicklungsbudget: CHF 5.0 / EUR 4 Mio.

Finanziell unabhängig und nachhaltig profitabel

Erfahrung aus mehr als 1'900 Projekten pro Jahr bei über 800 Kunden

**Technik allein bringt Sie nicht weiter.  
Man muss wissen, wie man sie richtig nutzt.**



5



2012 © Trivadis

BI Publisher Integration über Oracle Datenbanken  
19.06.2013

**trivadis**  
makes IT easier. ■ ■ ■

# AGENDA

1. BI Publisher Integration
2. BI Publisher Web Services
3. soapUI
4. Web Service Call-Outs der Datenbank
5. BI Publisher ReportService runReport
6. BI Publisher ScheduleService deliveryService
7. BI Publisher ScheduleService scheduleReport
8. Fazit



2012 © Trivadis

BI Publisher Integration über Oracle Datenbanken

19.06.2013

# BI Publisher Integration

Möglichkeiten einen Bericht auszuführen

- BI Publisher Server
- URL
- Java API
- Web Services

Integrationsmöglichkeiten für Applikationen

- Java API
- Web Services

Integrationsmöglichkeiten für Applikationen ohne Java

- Web Services



2012 © Trivadis

BI Publisher Integration über Oracle Datenbanken

19.06.2013

# Reporting Services

## Reporting-Lösung im Kundenprojekt

- APEX Applikation als Frontend
  - Zugriff auf Reporting Services
  - Eingabe der Parameter
  - Definition der Auswertungen
  - Speichern aller Eingaben als Service Request in der Datenbank
- Asynchrone Ausführung der Service Requests als Datenbank Job
  - Reportdaten lesen und speichern
  - Auswertungen erstellen
- Berichte mit BI Publisher
  - SQL Datenquellen
  - Template Builder für Word



2012 © Trivadis

BI Publisher Integration über Oracle Datenbanken

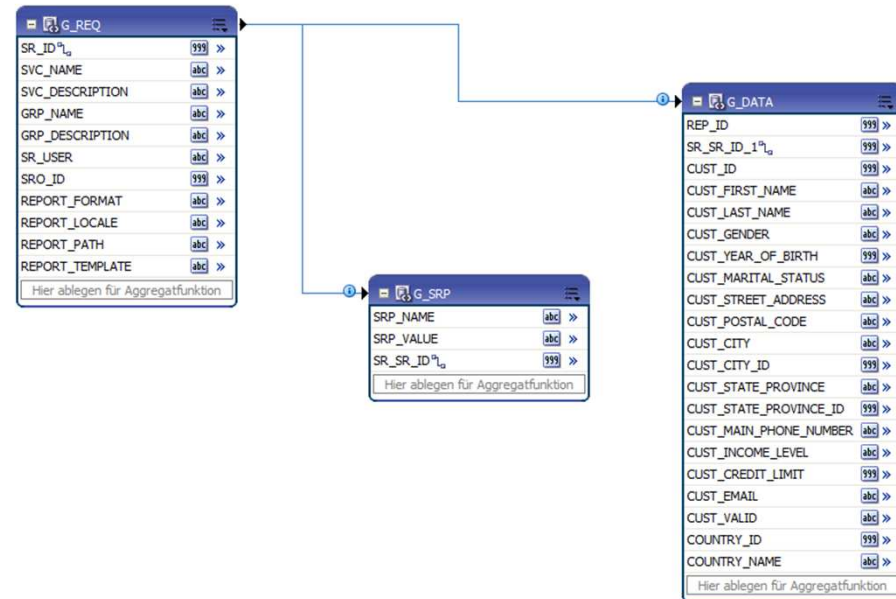
19.06.2013



# Reporting Services

## BI Publisher Datenmodell

- SQL Datenquellen
  - Service Request Daten
  - Service Request Parameter
  - Reportdaten
- Parameter
  - ID der Service Request Auswertung



# BI Publisher Web Services

## Übersicht der verfügbaren Web Services

- URL auf dem BI Publisher Server  
`http://<host>:<port>/xmlpserver/services`

## Dokumentation der Web Services

- Developer's Guide for Oracle BI Publisher

## Quelle für einen Einstieg

- Blog Eintrag Web Services in BI Publisher 11g

# BI Publisher Web Services

## BI Publisher 11g stellt neue Web Services zur Verfügung

- **CatalogService**
  - Methoden für die Verwaltung der Objekte im BI Publisher Katalog
    - Objekte sind Verzeichnisse, Berichte und Datenmodelle
  - *getFolderContents* erstellt eine Liste aller Objekte in einem Katalogverzeichnis
- **ReportService**
  - Methoden zur Ausführung von Berichten,
  - Methoden zur Abfrage von Metadaten zu Objekten
  - Methoden zur Verwaltung von Reportobjekten (Anlegen, Löschen)
  - *runReport* führt einen Bericht aus und gibt die erstellten Reportdaten zurückgibt

# BI Publisher Web Services

BI Publisher 11g stellt neue Web Services zur Verfügung

- **ScheduleService**
  - Methoden für den BI Publisher Scheduler
    - Aufsetzen von Report Jobs
    - Abholung von erzeugten Report Dokumenten
    - Abfrage von Job Status Informationen
  - *deliveryService* ermöglicht das Versenden der Reportdaten per E-Mail oder FTP
- **SecurityService**
  - Methoden für das BI Publisher Sicherheitsmodell
    - Verwaltung von Benutzern, Rollen und Berechtigungen für Katalogobjekte
    - An- und Abmelden am Server
  - *Login* erstellt einen SessionToken für den BI Publisher Server



# BI Publisher Web Services

Viele Methoden der Services sind in zwei Varianten verfügbar

- Stateless
  - Benutzernamen und ein Passwort wird bei jedem Aufruf benötigt
- Stateful "in-session"
  - Vorhandene Session wird verwendet
  - Anmeldung über den SecurityService
  - Methode *Login* generiert einen *bipSessionToken*
  - Aufruf aller "in-session" Methoden mit dem *bipSessionToken*

# soapUI



Testwerkzeug für SOAP-basierte Web Services

- Inspizieren
- Testen
- Ausführen

Download der frei verfügbaren Software

- <http://www.soapui.org/>

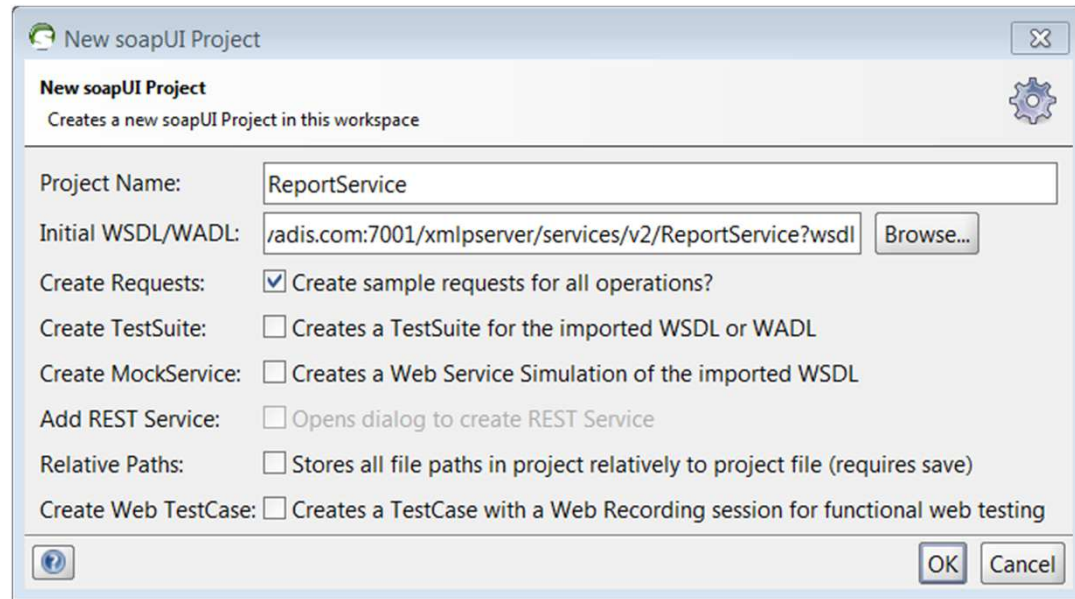


# soapUI Projekt



Neues Projekt für BI Publisher Web Service in soapUI anlegen

- Projektname
- WSDL Adresse



- Generiert für alle Methoden im WSDL Dokument SOAP Requests mit allen Parametern

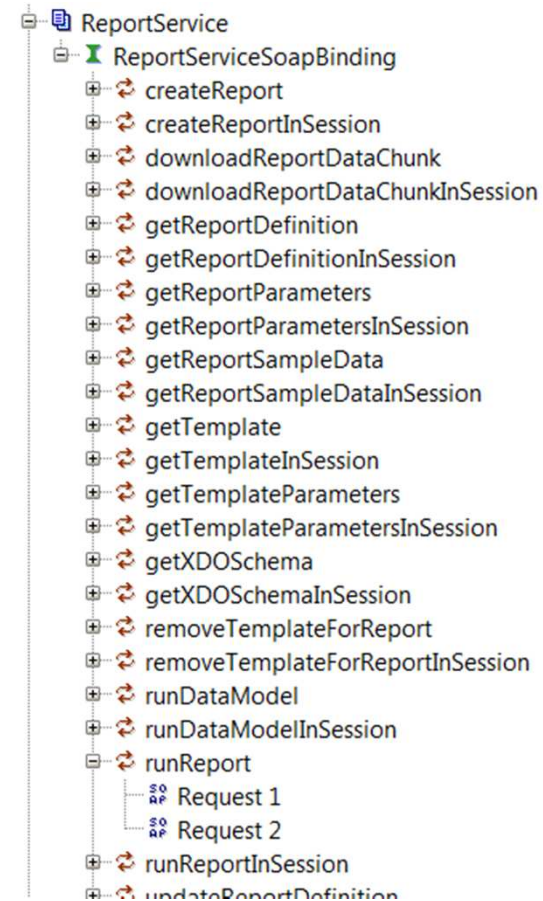


# soapUI generierte Methoden

- Generierte Methoden für BI Publisher Reportservice
- Request für Methode runReport



SoapUI





# soapUI Request



SoapUI

## runReport Request

- XML Dokument
  - SOAP Envelope mit Namespace
  - SOAP Header
  - SOAP Body
  - Parameter

## runReport Response

- XML Dokument
  - SOAP Envelope mit Namespace
  - SOAP Body mit Report als Byte-Array

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v2="http://xmlns.oracle.com/oxp/service/v2">
  <soapenv:Header/>
  <soapenv:Body>
    <v2:runReport>
      <v2:reportRequest>
        <v2:XDOPropertyList>
          <v2:metaDataList>
            <!--Zero or more repetitions:-->
            <v2:item>
              <v2:metaDataName?</v2:metaDataName>
              <v2:metaDataValue?</v2:metaDataValue>
            </v2:item>
          </v2:metaDataList>
        </v2:XDOPropertyList>
        <v2:attributeCalendar?</v2:attributeCalendar>
        <v2:attributeFormat?</v2:attributeFormat>
        <v2:attributeLocale?</v2:attributeLocale>
        <v2:attributeTemplate?</v2:attributeTemplate>
        <v2:attributeTimezone?</v2:attributeTimezone>
        <v2:byPassCache?</v2:byPassCache>
        <v2:dynamicDataSource>
      </v2:reportRequest>
    </v2:runReport>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <runReportResponse xmlns="http://xmlns.oracle.com/oxp/service/v2">
      <runReportReturn>
        <metaDataList xsi:nil="true"/>
        <reportBytes>JVBERi0xLjYNCjYgMCAvYmoNCjw8IC9MZW5ndGggNTAxIC9GaWw0ZXIgLg</reportBytes>
        <reportContentType>application/pdf</reportContentType>
        <reportFileID xsi:nil="true"/>
        <reportLocale xsi:nil="true"/>
      </runReportReturn>
    </runReportResponse>
  </soapenv:Body>
</soapenv:Envelope>
```



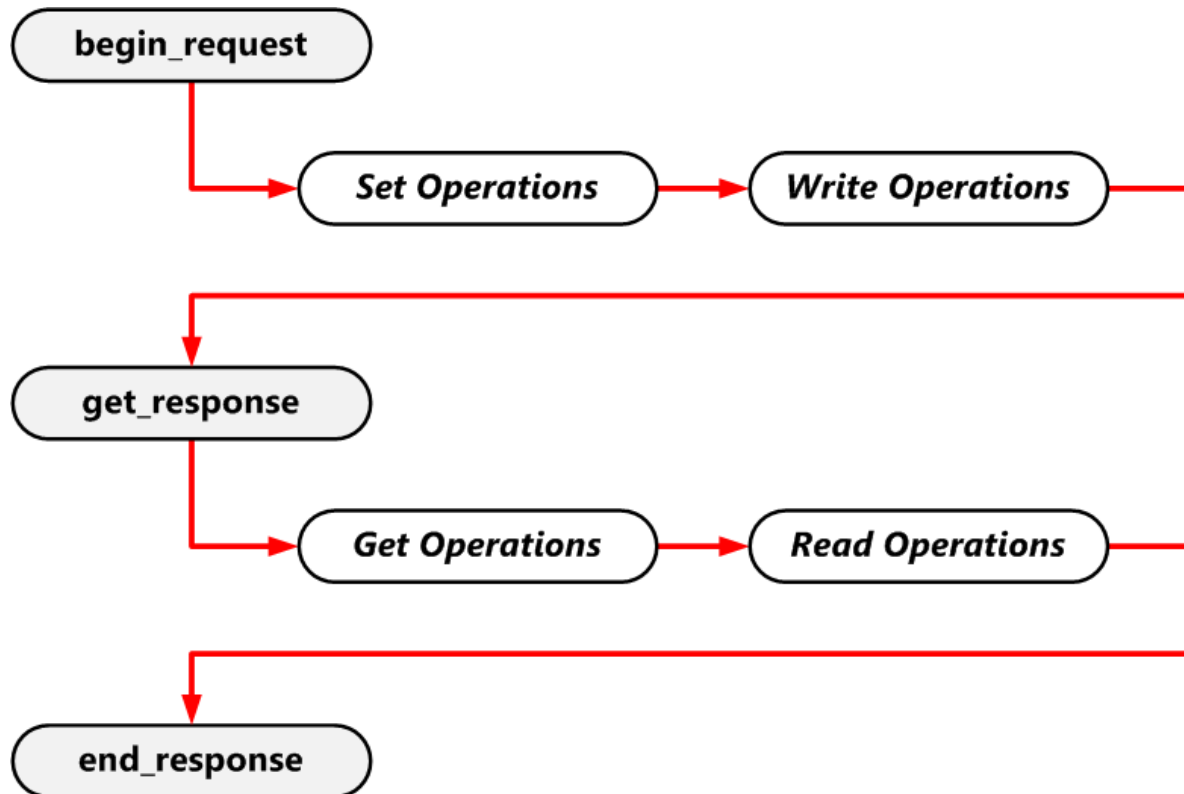
# Web Service Call-Outs der Datenbank

Die Oracle Datenbank 11g bietet zwei PL/SQL Lösungen, um Web Services in der Datenbank zu verarbeiten:

- *UTL\_DBWS*
  - Verfügbar ab Oracle 10g
  - Wrapper Package für JAX-RPC Java Clients generiert mit JPublisher
  - Ermöglicht direkte SOAP Aufrufe
  - Das Package erfordert die Installation zusätzlicher Java Klassen in der Datenbank (JPublisher)
- *UTL\_HTTP* ist
  - Verfügbar ab Oracle 7.3.4
  - Bietet die Möglichkeit HTTP Aufrufe direkt aus der Datenbank durchzuführen
  - Das Package kommuniziert mit einem Web Service über SOAP Messages

# Web Service Call-Outs mit UTL\_HTTP

Schematischer Request-Response-Ablauf eines Call-Outs mit UTL\_HTTP



# Web Service Call-Outs mit UTL\_HTTP

- begin\_request
  - Set Operationen
    - HTTP Header Informationen setzen
  - Write Operationen
    - Request schreiben
- get\_response
  - Get Operationen
    - Status Informationen holen
  - Read Operationen
    - Response lesen
- end\_response
  - Request beenden

# Datenbank Berechtigungen

## Netzwerk-Berechtigungen

- Der ausführende Datenbank-User benötigt Netzwerk-Berechtigungen für den Host und den Port des BI Publisher Servers
- Datenbank-Administrator mit SYSDBA Privilegien erteilt Berechtigungen
- Access-Control-List anlegen
  - DBMS\_NETWORK\_ACL\_ADMIN.CREATE\_ACL
- CONNECT Privilegien für Host und Port des BI Publisher Servers zur Access-Control-List hinzufügen
  - DBMS\_NETWORK\_ACL\_ADMIN.ADD\_PRIVILEGE

# BI Publisher ReportService runReport

Report ausführen  
Web Service ReportService

- Wichtige Felder der Methode *runReport*
  - ReportRequest
    - ParamNameValues
  - userID
  - password

Für die Stateless Methode sind die Parameter userID und password eines BI Publisher Users anzugeben

# SOAP Message ReportRequest

## Attribute ReportRequest

- attributeFormat
  - BI Publisher Formatangabe
  - Wert: pdf
- attributeLocale
  - Bestimmt regionale Einstellungen (Datum, Zeit, Sprache)
  - Wert: de\_DE
- attributeTemplate
  - Name der Vorlagendatei, nicht der logische Name der Vorlage
  - Wert: customers1.rtf

# SOAP Message ReportRequest

## Attribute ReportRequest

Die Parameterangaben beziehen sich auf die Parameter des Datenmodells

- parameterNameValues/listOfParamNameValues/item/dataType
  - Datentyp des Parameters
  - Wert: xsd:int
- parameterNameValues/listOfParamNameValues/item/name
  - Name des Parameters
  - Wert: p\_sro\_id
- parameterNameValues/listOfParamNameValues/item/values/item
  - Wert des Parameters
  - Wert: 1



# SOAP Message ReportRequest

## Attribute ReportRequest

- reportAbsolutePath
  - Vollständige Pfadangabe im BI Publisher Katalog zum Report
  - Wert: /ReportingServices/customers.xdo
- sizeOfDataChunkDownload
  - Bestimmt, wie die erzeugten Reportdaten zurückgegeben werden.
  - Wert: -1 (transferiert die Daten vollständig)

# PL/SQL Package für ReportService

Eigenes PL/SQL Package für Web Service Methode *runReport* als Schnittstelle zwischen Applikation und BI Publisher.

Prozedur `request_report`

- Aufrufschnittstelle der Applikation
- Identifiziert die geforderte Auswertung
- Aufbereitung für Aufruf Web Service Call-Out
- Verarbeitung der Response
- Fehlerbehandlung
- Statusinformation der Auswertung

# PL/SQL Package für ReportService

Eigenes PL/SQL Package für Web Service Methode *runReport*  
Schnittstelle zwischen Applikation und BI Publisher.

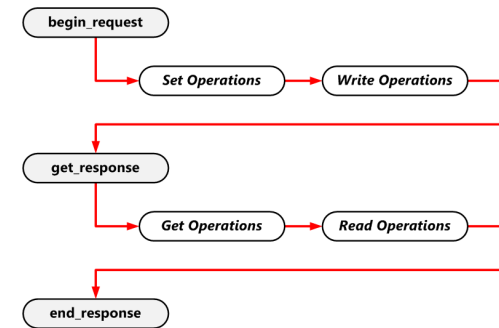
Funktion run\_report

- Aufrufschnittstelle BI Publisher
- Input Parameter reportrequest als Object Type
- Web Service Request/Response Handling mit UTL\_HTTP
- SOAP Message
- Return Wert ist die Web Service Response als CLOB

# Request-Response Verarbeitung mit UTL\_HTTP

## Besonderheiten

- `begin_request` Set Operationen
  - Transfer Timeout setzen, Default 60 Sekunden
  - HTTP Header `Content-Length` ist die Länge der SOAP Message
  - Body Charset ist UTF-8
  - HTTP Header `SOAPAction` ist `runReport`
- `begin_request` Write Operationen
  - SOAP Message mit `UTL_RAW` in Raw-Wert konvertieren
  - Verwende `write_raw` um Sonderzeichen richtig darzustellen
- `get_response` Read Operationen
  - `read_text` mit Länge 32767
  - Schreiben in CLOB



# BI Publisher ScheduleService deliveryService

Versenden der Reportergebnisse über verschiedene Kanäle  
Web Service ScheduleService

- Wichtige Felder der Methode *deliveryService*
  - DeliveryRequest
    - DeliveryChannels
    - BIPDataSource
  - userID
  - password
- Versandoptionen (deliveryChannels) der Methode *deliveryService*
  - E-Mail, Drucker, Fax, FTP, HTTP, WebDav
- Für die Versandoptionen sind entsprechende BI Publisher Zustellungskonfigurationen erforderlich

# BI Publisher Zustellungskonfiguration

## E-Mail

BI Publisher Administration  
Zustellung E-Mail  
E-Mail Server

- Servername                      Logischer Name des Mail Servers
- Host                                Hostname des Mail Servers
- Port                                 Port des Mail Servers
- Sichere Verbindung
- Benutzername
- Kennwort

# BI Publisher Zustellungskonfiguration FTP

BI Publisher Administration  
Zustellung FTP  
FTP Server



- Servername Logischer FTP Server Name
- Host Hostname im Netzwerk
- Port Port für FTP Zugang
- Secure FTP verwenden Nur aktivieren, wenn SFTP verwendet wird
- Benutzername Benutzer auf dem Host
- Kennwort Kennwort für den Benutzer auf dem Host

# PL/SQL Package für Versendung

Eigenes Package für die Versendung

Spezielle Prozeduren für die implementierten Versandoptionen

- Speichern in der Datenbank
- Versenden per E-Mail
- Versenden per FTP

Die Reportdaten aus der Methode *runReport* werden verarbeitet.





# SOAP Message bei Versendung

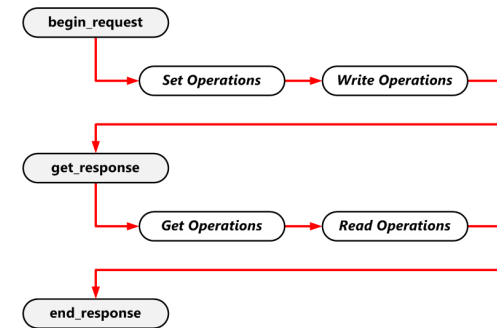
## Besonderheiten

- `contentType`
  - Das Reportdokument wird als Attachment an die Mail angehängt
  - Der Dateiname und die Dateierweiterung werden automatisch vergeben
- `emailOptions/emailServerName`
  - Logischer Name der E-Mail Zustellungskonfiguration
- `ftpOptions/ftpServerName`
  - Logischer Name der FTP Zustellungskonfiguration
- `dynamicDataSourcePath`
  - Vorgegebener Dateiname
  - Übersteuert die Automatik durch Parameter `contentType`

# Request-Response Verarbeitung mit UTL\_HTTP

## Besonderheiten

- `begin_request` Set Operationen
  - HTTP Header Transfer-Encoding "chunked"
- `begin_request` Write Operationen
  - Chunk Size 1024
  - Keine Content-Length
  - Verwende `write_raw`
- `get_response` Read Operationen
  - `read_text` mit Länge 32767
  - Schreiben in CLOB



# BI Publisher ScheduleService scheduleReport

Definition eines Scheduler Jobs  
Web Service ScheduleService

- Wichtige Felder der Methode *scheduleReport*
  - ReportRequest
  - DeliveryChannels
  - und viele mehr
- Erstellt einen Scheduler Job und gibt die Job ID zurück
- Scheduler Job
  - Asynchron Ausführung auf dem BI Publisher Server
  - Report wird ausgeführt
  - Reportergebnis wird versendet
- Kontrolle der Jobs über Job Historie (*getAllScheduledReportHistory*)



# Fazit

## Vorteile

- Schnittstelle für Applikationen mit Oracle Datenbank
  - APEX
  - Forms
- Kontrolle der Web Service Verarbeitung
- Stabile Ausführung der Web Services

## Nachteile

- Umsetzung der Schnittstelle in der Datenbank ist aufwendig

# Referenzen



## Referenzen

- **Developer's Guide for Oracle Business Intelligence Publisher**

Oracle® Fusion Middleware Documentation Library 11g Release 1 (11.1.1.6.0)

[http://docs.oracle.com/cd/E23943\\_01/bi.1111/e22259/webservices.htm#T569886](http://docs.oracle.com/cd/E23943_01/bi.1111/e22259/webservices.htm#T569886)

- **Web Services in BI Publisher 11g**

Rittman Mead Consulting, Blog Archive, Robin Moffatt 2011

<http://www.rittmanmead.com/2011/11/web-services-in-bi-publisher-11g>

- **Implementierung von Web Services in Oracle-Datenbankanwendungen**

Florin Serban, pitss, Februar 2011

[http://www.pitss.de/fileadmin/pitss/images/de/White\\_Papers/WhitePaper\\_WebServices\\_DE.PDF](http://www.pitss.de/fileadmin/pitss/images/de/White_Papers/WhitePaper_WebServices_DE.PDF)

VIELEN DANK.

Trivadis AG

Perry Pakull

Europa-Strasse 5  
CH-8152 Glattbrugg

Tel. +41-44-808 70 20

Fax +41-44-808 70 21

[info@trivadis.com](mailto:info@trivadis.com)

[www.trivadis.com](http://www.trivadis.com)

BASEL

BERN

LAUSANNE

ZÜRICH

DÜSSELDORF

FRANKFURT A.M.

FREIBURG I.BR.

HAMBURG

MÜNCHEN

STUTTGART

WIEN

39

2012 © Trivadis

BI Publisher Integration über Oracle Datenbanken  
19.06.2013

**trivadis**  
makes IT easier. ■ ■ ■