

Effektive Dokumentenerstellung durch Modularität mit dem BI Publisher

Joachim Metzdorf
BASF IT Services GmbH
Ludwigshafen

Rainer Willems
Oracle Deutschland B.V. & Co. KG
Geschäftsstelle Frankfurt

Schlüsselworte:

BI Publisher, Subtemplates, Webservices, Erfahrungsbericht

Einleitung

Hinter der europaweiten Neuausrichtung der Personalbeschaffung der BASF steht das Projekt Zeppelin des Bereiches Human Resources. Ziel des Projektes ist, die Rekrutierung der BASF europaweit neu auszurichten, sowie professioneller und effektiver zu gestalten. Dabei sollen die Rekrutierungsprozesse transparenter und zügiger verlaufen, sowie besser auf die individuellen Bedürfnisse abgestimmt werden.

Teil des Projektes ist ein Correspondence-Tool (Schriftverkehrsmanagement), mit dem aus SAP ca. 170 Dokumente wie Bescheinigungen, Arbeitsverträge und Anschreiben erstellt werden. Dieses Tool löst eine auf MS-Word-Serienbrief und OLE2 basierte Anwendung ab. Die Hauptanforderung des neuen Tools war, dass eine rein webbasierte Anwendung erstellt wird, sowohl für die SAP-eRecruiting-Anwendung, als auch für das verwendete Berichtswerkzeug. Damit kann das Rendering der Berichte serverseitig ausgeführt werden und das SAP-seitige Lesen der Daten kann auf dynamischen Methoden basieren, was durch Customizing gesteuert werden kann.

In einer Evaluierungsphase wurde festgestellt, dass diese Anforderungen mit dem BI Publisher erfüllt und die gewünschten Berichte im rtf-Format generiert werden können.

BI Publisher – Eine kurze Einführung

Der Oracle BI Publisher ist eine Reporting-Lösung, die unterschiedlichste Datenquellen verwenden kann, um daraus Dokumente zu erzeugen. Hierbei wird die Trennung von Daten, Layout und Sprache konsequent betrieben, um dadurch an Flexibilität und Wartbarkeit der Lösungen zu gewinnen.

Im Unterschied zu klassischen Berichtswerkzeugen werden im BI Publisher die Informationen zu Daten, Layout und Sprachen getrennt voneinander gehalten und erst zur Laufzeit zusammengebracht. Dies erhöht deutlich die Flexibilität, da beispielsweise zu einer Datendefinition mehrere Layoutvorlagen und Sprachsets zugeordnet werden können.

Der BI Publisher verarbeitet XML-Daten und hilft, diese aus relationalen oder multidimensionalen Datenbanken, verschiedenen Lösungen oder der BI EE zu erzeugen. Die XML-Erzeugung kann aber auch – wie in diesem Projekt – ausgelagert werden.

Das Layout wird durch XSL-FO beschrieben. Auch hierbei gibt es die Unterstützung zur Generierung dieser Sprache aus diversen Werkzeugen wie Microsoft Word (Open Office in Planung), Adobe Professional oder ab Release 11g auch mit einer browserbasierten Oberfläche.

Falls Mehrsprachigkeit gewünscht ist, kann diese dynamisch mittels XLIFF hinzugefügt werden.

Durch die unterschiedlichen Möglichkeiten des Aufrufes und der Zielformate (in diesem Projekt wurden WebServices verwendet) kann der BI Publisher gut in heterogene Landschaften mit verschiedenen Anforderungen integriert werden.

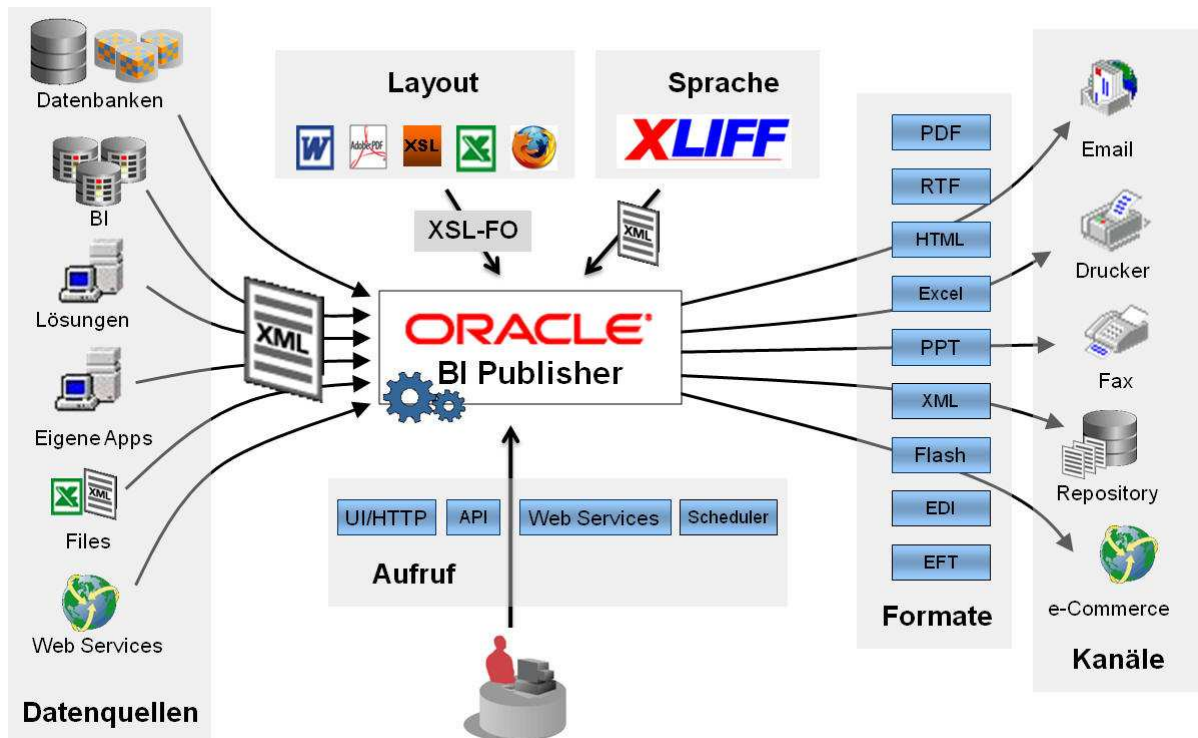


Abb. 1: Architektur des BI Publishers

Von SAP zum BI Publisher und zurück

Das Schriftverkehrsmanagement ist eine SAP Web Dynpro ABAP-Anwendung, mit der in mehreren Schritten Kandidaten, Prozesse und zu erstellende Dokumente ausgewählt, sowie die ggf. notwendigen Berichtsdaten prozessbezogen in einem Fragebogen manuell eingegeben werden können.

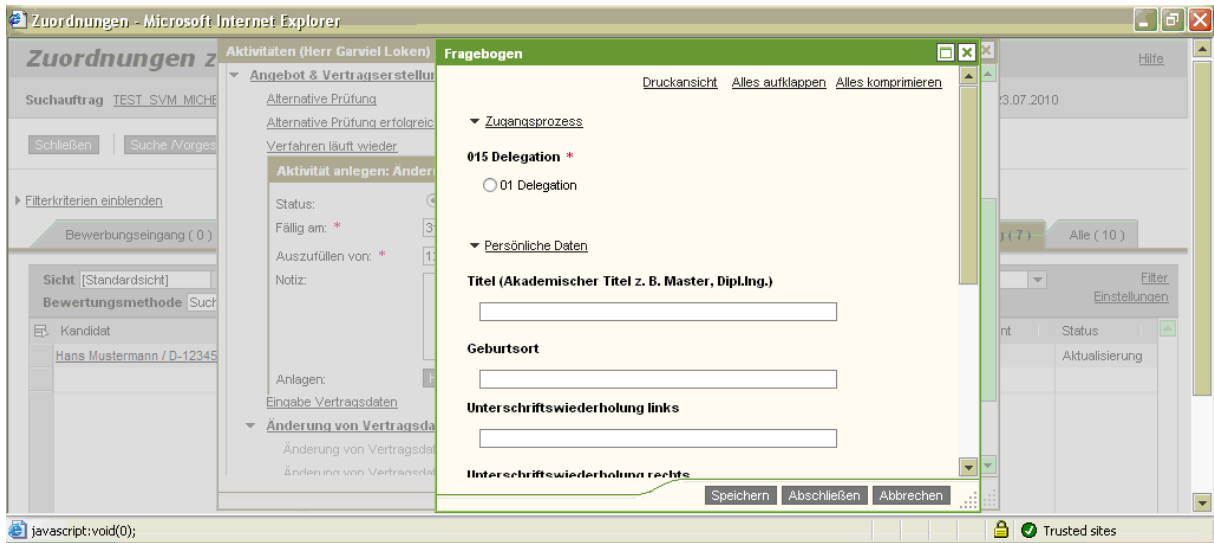


Abb. 2: Vorbereitung Berichtsgenerierung in SVM

Danach erfolgt die Generierung der zum Prozess gehörenden Dokumente durch Aufruf des BI Publishers via Webservice. Die vom BI Publisher erstellten Dokumente werden SAP-seitig abgelegt und können in der Anwendung geöffnet, falls erforderlich geändert und per Mail verschickt werden.

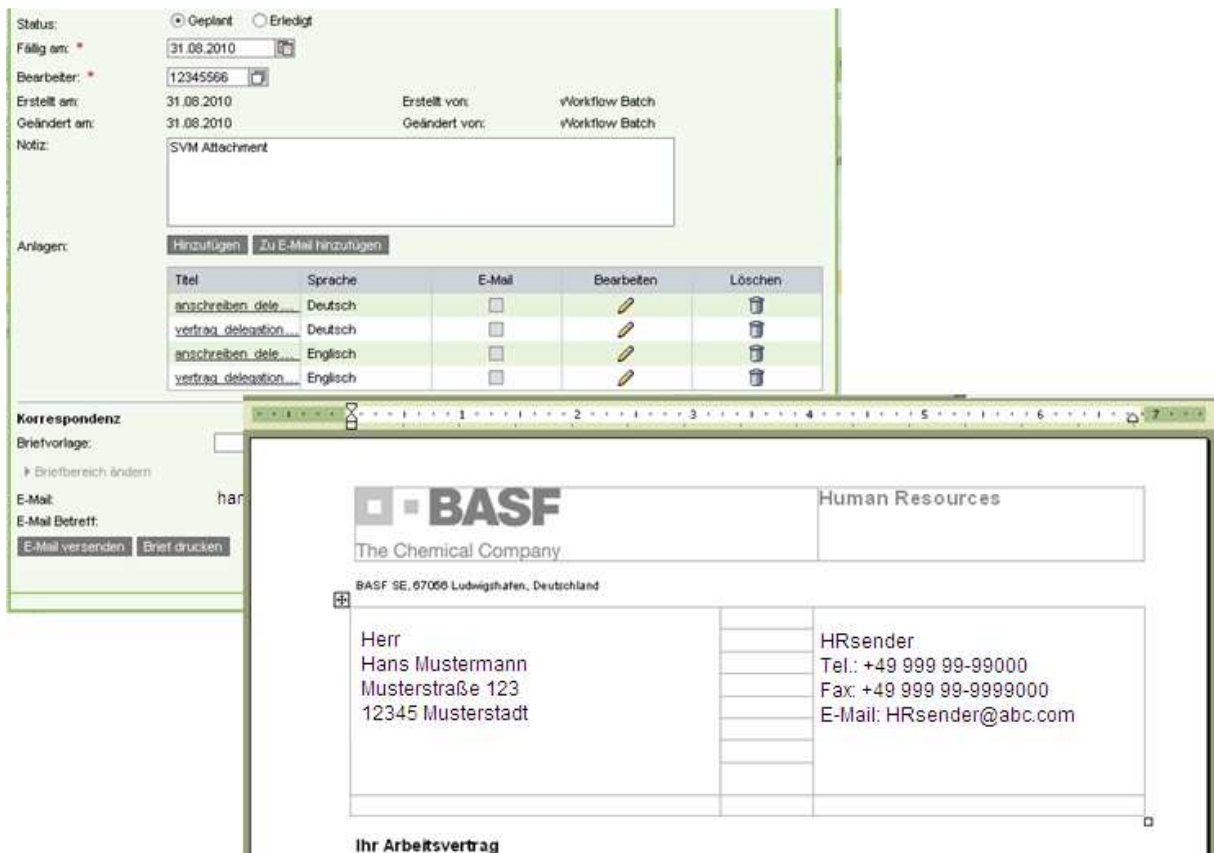


Abb. 3: Erstellte Dokumente in SVM

Kernstück der SAP-seitigen Realisierung sind 2 in ABAP programmierte Funktionsbausteine. Dabei arbeitet die äußere Funktion alle vom Benutzer eingegebenen Informationen in einer Schleife ab. Die innere Funktion übernimmt die Vorbereitung, BI Publisher-Generierung und Nachbereitung jedes einzelnen Dokumentes vor. Zunächst werden die notwendigen Berichtsdaten aus SAP-Tabellen gelesen und die XML-Daten über eine Subfunktion als Textstring generiert. Danach werden diese Daten mit einer weiteren Subfunktion base64-codiert. Diese Codierung wird für die Datenübergabe vom Webserviceparameter „reportData“ vorausgesetzt. Danach erfolgt der eigentliche Webserviceaufruf zur Generierung der Berichtsausgabe im rtf-Format durch einen SOAP-Request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:pub="http://xmlns.oracle.com/oxp/service/v11/PublicReportService"
  xmlns:pub1="http://xmlns.oracle.com/oxp/service/PublicReportService" >
  <soapenv:Header/>
  <soapenv:Body>
    <pub:runReport>
      <pub:reportRequest>
        <pub:attributeCalendar/>
        <pub:attributeFormat>rtf</pub:attributeFormat>
        <pub:attributeLocale>de</pub:attributeLocale>
        <pub:attributeTemplate/>
        <pub:flattenXML>>false</pub:flattenXML>
        <pub:reportAbsolutePath>/zeppelin/prozess_a/anschreiben/
          anschreiben.xdo</pub:reportAbsolutePath>
        <pub:reportData>77u/PD94bWwgdMvyc2lvbj0iMS4wIiBlbmNvZGlucy0idXRm
          LTE2Ij8+CjxST1dTRVQ+PFJpVz48QU5SRUQ+SGVycjwvQU5SRUQ+PEFVU1dTPjwv
          QVVTV1M+PEFVU1dUPjQVVTV1Q+PEFVU1dVPjwvQVVTV1U+PEJFVFEJOPjU1NTU1PC
          ...
          J1Y2h0czwvVU5UUkU+PFZPUk5BPkYydWN1PC9WT1JOQT48W1VHQU4+MDEyPC9aVU
          dBTj48U1BSU0w+RDwwU1BSU0w+PC9ST1c+PC9ST1dTRVQ+</pub:reportData>
        <pub:sizeofDataChunkDownload>-1</pub:sizeofDataChunkDownload>
      </pub:reportRequest>
      <pub:userID>username</pub:userID>
      <pub:password>mypassword</pub:password>
    </pub:runReport>
  </soapenv:Body>
</soapenv:Envelope>
```

Abb. 4: BI Publisher SOAP-Request

Der Webservice liefert einen base64-codierten Bytestring zurück, der wiederum durch eine Subfunktion decodiert werden muss. Das Ergebnis wird dann in einer SAP-Tabelle binär abgelegt und kann als rtf-Dokument mit SAP-Funktionalität geöffnet werden.

Für den Aufruf des BI Publishers mussten die aus der BI Publisher-WSDL in ABAP generierten Klassen manuell erweitert werden, da die automatisch erstellten Klassen keine Attachments ermöglichen, dies aber für die Übergabe der Berichtsdaten unabdingbar war.

Erstellung Berichtsvorlagen mit dem BI Publisher (MS-Word-Client)

Wegen der großen Anzahl der zu erstellenden Berichte wurde bei Projektbeginn ein Regelwerk erarbeitet, um den Kommunikationsaufwand zwischen Fachabteilung und BI Publisher-Entwicklung gering zu halten. Da die Dokumente im VORSYSTEM bereits vorhanden waren, wurden diese seitens der Fachabteilung anhand des Regelwerks so überarbeitet, dass die Entwickler ohne mündliche Kommunikation die notwendigen Datenfelder und ggf. Logik einarbeiten konnten. Dazu wurde an den Stellen der Berichte, an denen Dateninhalte platziert werden, der Name des Datenfeldes eingefasst in # eingetragen.

Sehr #ANRED_FORM# #ANRED# #NAME#
Anbei erhalten Sie den Vertrag für ...

Abb. 5: Template-Vorlagen (Felder)

Um Logik abbilden zu können, wurde wie folgt verfahren:

Bedingung **SONST**:
WENN #FELD1# = X DANN Text: Sie haben x Monate
Probezeit SONST Text: Sie haben y Monate Probezeit

Bedingung **ohne SONST**:
WENN #MATYP# = 1 dann Text: Gesundheitszeugnis
ist erforderlich

Abb. 6: Template-Vorlagen (Logik)

Durch die Entwickler wurden anschließend die Felder und die Logik in die endgültigen Berichtsvorlagen eingearbeitet.

Um Berichtsbausteine wie Textpassagen, Logos, Header, Footer usw. auszulagern, wurde die BI Publisher Funktionalität der Subtemplates verwendet.

The image shows a report template layout with several text blocks and logos. The blocks are: BASF logo (The Chemical Company), BASF SE address, HRsender contact information, and a salutation 'Ihr Arbeitsvertrag' followed by 'Sehr geehrter Herr Mustermann'.

Abb. 7: Template-Vorlagen (Textbausteine)

Textbausteine des Subtemplates können wiederum Grafiken, Tabellen, sowie Dateninhalte ggf mit Logik beinhalten.

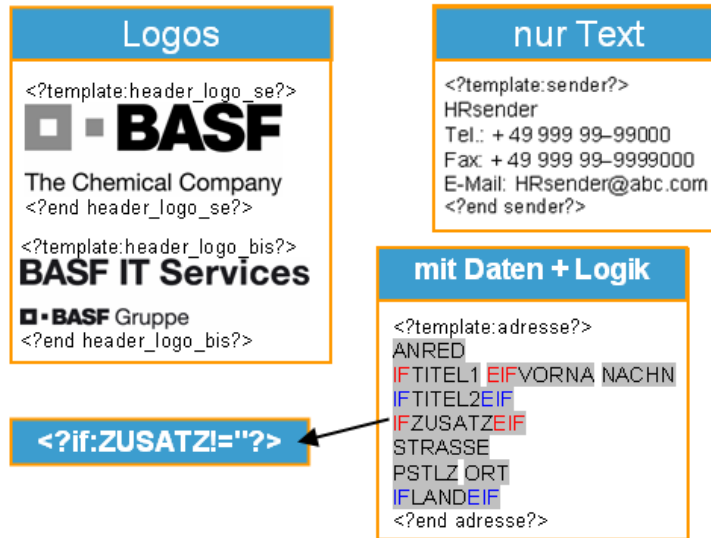


Abb. 8: SubTemplate (Textbausteine)

Die Verweis auf das Subtemplate erfolgt im Bericht durch eine Import-Anweisung an erster Stelle der Berichtsvorlage.

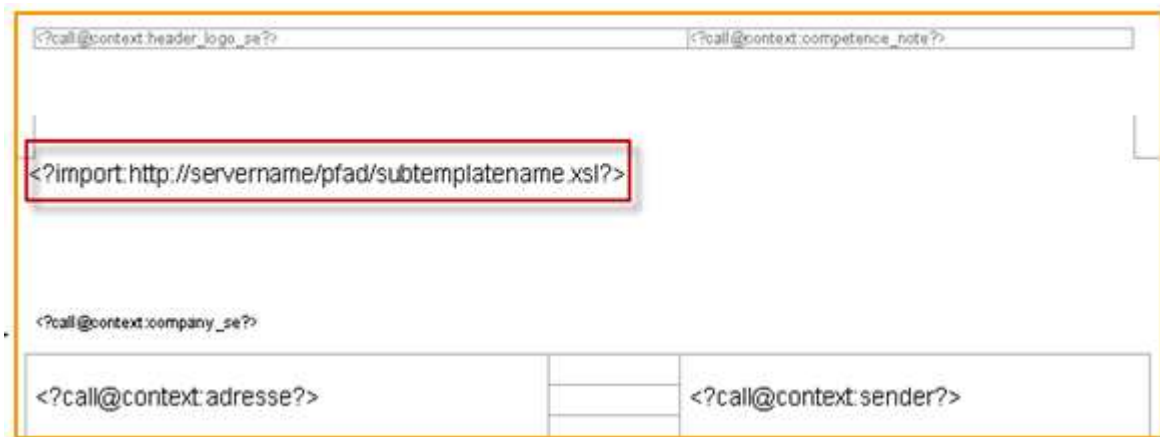


Abb. 9: SubTemplate aufrufen

Berichterstellung mit dem BI Publisher (Server)

Da die Berichte ihre Daten ausschließlich über den Webserviceaufruf im XML-Format erhalten, wurden die Berichte auf dem BI Publisher-Server ohne Datenquelle eingerichtet. Für einen Bericht musste lediglich ein Layout erstellt und die Berichtsvorlage hochgeladen werden.

Nachteil: Die Lauffähigkeit konnte im BI Publisher nicht direkt getestet werden. Die Tests erfolgten deshalb anhand von Beispieldaten im XML-Format direkt in Word.

Realisierungsprobleme

Im Zuge der Entwicklungsarbeiten sind folgende Probleme aufgetreten:

- Automatische serverseitige Konvertierung des Subtemplates unter Tomcat 5.5. wurde mit Fehler abgebrochen
 - ⇒ Subtemplate-Export in MS-Word (Exportieren > XSL-FO-Stylesheet) manuell vornehmen
- Zusätzliche Leerzeile in generierten rtf-Outputs nach jedem Subtemplate-Aufruf
 - ⇒ Durch BI Publisher Bugfix April 2010 behoben
- Zusätzliche Leerzeile in generierten rtf-Outputs nach Aufruf von Subtemplate-Tabellen (z.B. Footer)
 - ⇒ Entfernen des überflüssigen XSL-FO-Blockes mittels Korrekturprogramm (Eigenentwicklung)
- Automatisch generierte Einrückungen in Subtemplate-Aufrufen (5.4pt)
 - ⇒ Replace der Einrückungen von 5.4pt auf 0.0pt mittels Korrekturprogramm (Eigenentwicklung)
- Blocksatz links-/rechtsbündig in Aufzählungen wird als linksbündiger Flattersatz generiert
 - ⇒ Die Problemlösung ist zurzeit noch in Arbeit

Erfahrungen

Mit dem BI Publisher konnten in diesem Projekt viele Berichte effizient erstellt werden, insbesondere durch

- Wiederverwendbarkeit bestehender Berichtsvorlagen im Corporate Design
- Aufgabenteilung Facheinheit / Entwicklung bei Templateerstellung
- Minimaler Kommunikation Facheinheit / Entwicklung (durch Regelwerk)
- Auslagerung von Textbausteinen in Subtemplates
- Layoutänderungen können durch Subtemplates schnell umgesetzt und deployed werden.
- Sehr geringer Aufwand für die serverseitige Berichtserstellung
- Geringer Lernaufwand für Entwickler

Im Betrieb läuft der BI Publisher sehr stabil auf kostengünstiger Standard-Hardware; die Betreuung des BI Publisher-Servers wird von der Webserverbetreuung im Routinebetrieb übernommen.

Fazit

Durch die Unabhängigkeit der Entwicklungsumgebung (MS-Word) bietet der BI Publisher eine sehr effiziente Möglichkeit, Massenberichte zu erstellen. Insbesondere dann, wenn die Vorlagen in den Fachabteilungen bereits als Word-Dokument vorhanden sind. Der Aufruf kann aus jeder beliebigen (J2EE-)Anwendung über Webservice erfolgen. Die leider aufgetretenen Probleme in rtf-Outputs konnten durch Bugfixes behoben bzw. durch Workarounds umgangen werden, schmälern daher den positiven Gesamteindruck nicht, zumal die realisierte SAP - BI Publisher-Schnittstelle in weiteren Projekten wiederverwendet werden kann.

Kontaktadressen:

Joachim Metzdorf

BASF IT Services GmbH
 Jaegerstraße 1
 D-67059 Ludwigshafen

Telefon: +49 (0) 621-60-49084
 Fax: +49 (0) 621-60-95250
 E-Mail: joachim.metzdorf@basf-it-services.com
 Internet: www.basf-it-services.com

Rainer Willems

Oracle Deutschland B.V. & Co. KG
Robert-Bosch-Str. 5
D-63303 Dreieich

Telefon: +49 (0) 6103-397-290
Fax: +49 (0) 6103-397-111
E-Mail: rainer.willems@oracle.com
Internet: www.oracle.com/de