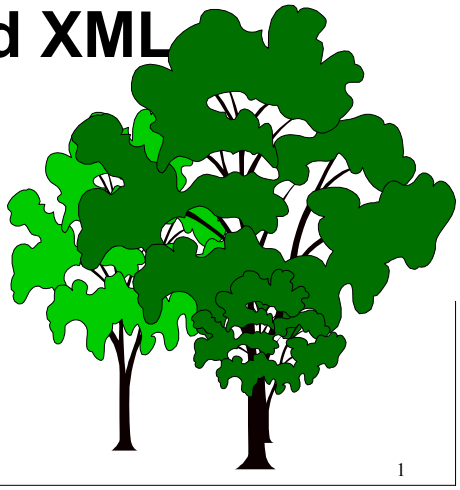
Orientation in Objects

JBuilder 5 und XML



Michael Schmut
Thomas Forster

schmut@oio.de

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de

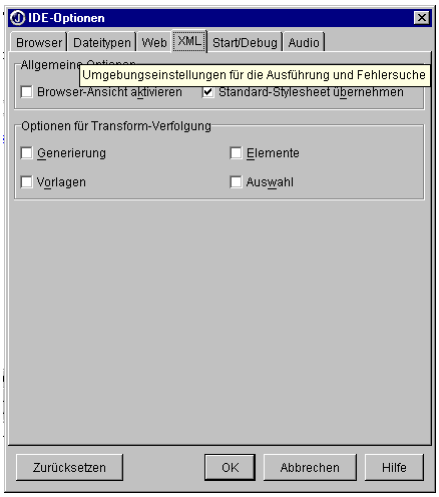
Inhalt

Orientation in Objects

- XML - Konfiguration von JBuilder
- XML - Umwandlung und Validierung
- Web - Publishing mit Cocoon
- XML - Databinding mit Castor und BorlandXML
- XML - Datenbankkomponenten im JBuilder

© 2000-2001 Orientation in Objects GmbH

XML - Konfiguration von JBuilder



- **Browseransicht aktivieren :**
  macht den integrierten Browser verfügbar
- **Standardstylesheet übernehmen :**
  zeigt die Baumansicht einer XML -Datei
- **Transform - Verfolgung :**
  Optionen für die Transform-Verfolgung mit deren Hilfe bei einer Umwandlung der Programmfluß im Meldungsfenster verfolgt werden kann , wenn ein Stylesheet angewendet wird.

© 2000-2001 Orientation in Objects GmbH

Inhalt

- **XML - Konfiguration von JBuilder**
- **XML - Umwandlung und Validierung**
- **Web - Publishing mit Cocoon**
- **XML - Databinding mit Castor und BorlandXML**
- **XML - Datenbankkomponenten im JBuilder**

© 2000-2001 Orientation in Objects GmbH

XML – Umwandlung und Validierung

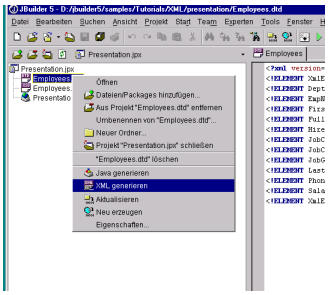
Orientation in Objects

- Integration des Xerces - Parsermoduls
- DTD zu XML Experte
- XML zu DTD Experte
- Validierung gegen Schemata und DTDs
- XML – Betrachter (Browser)

© 2000-2001 Orientation in Objects GmbH

Erstellen eines XML - Files aus einer DTD

Orientation in Objects

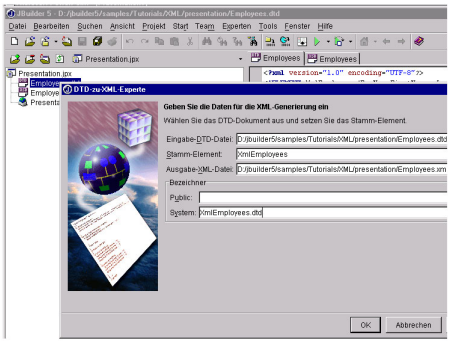


- Rechtsklick auf die DTD
- „XML - Erstellen“ auswählen

© 2000-2001 Orientation in Objects GmbH

Erstellen eines XML - Files aus einer DTD

Orientation in Objects



- Root - Element auswählen
- Ausgabe - Datei angeben
- im Feld „System“ die zu referenzierende DTD angeben

© 2000-2001 Orientation in Objects GmbH

Inhalt

Orientation in Objects

- **XML - Konfiguration von JBuilder**
- **XML - Umwandlung und Validierung**
- **Web - Publishing mit Cocoon**
- **XML - Databinding mit Castor und BorlandXML**
- **XML - Datenbankkomponenten im JBuilder**

© 2000-2001 Orientation in Objects GmbH

Probleme beim Web - Publishing

Orientation in Objects

Webpublishing

- Starres Layout
- Großer Änderungsaufwand
- Aufwand im Contentmangement

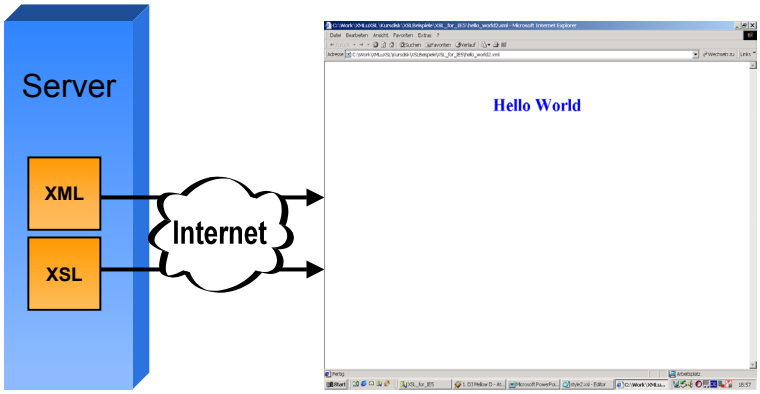
Webentwicklung

- Unübersichtliche ServerPage Anwendungen
- Problematische Trennung der Kompetenzen und Aufgaben
- Starres Layout

© 2000-2001 Orientation in Objects GmbH

XSL im Browser

Orientation in Objects



© 2000-2001 Orientation in Objects GmbH

Serverside XML

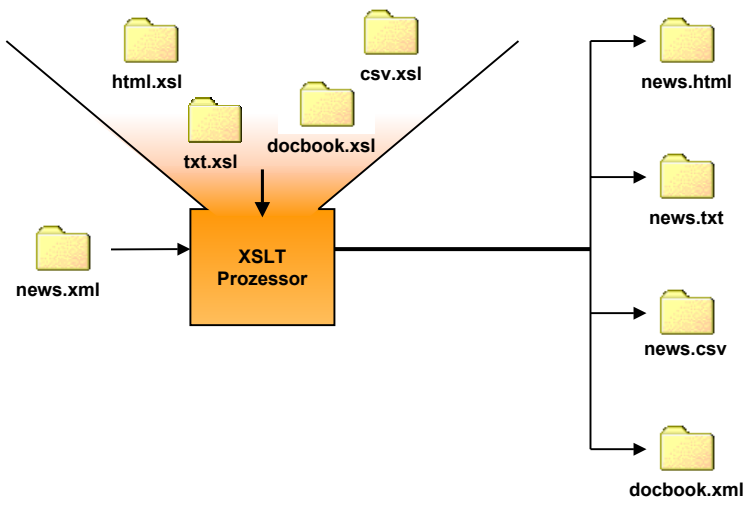
Orientation in Objects

- Server wandelt XML in HTML, WML,... um
- Aufbereitung für verschiedene Clients ist möglich
  - IE, Netscape, WAP-Client, ...
- Keine speziellen Anforderungen an den Browser
  - Java, Neue Version, ...

© 2000-2001 Orientation in Objects GmbH

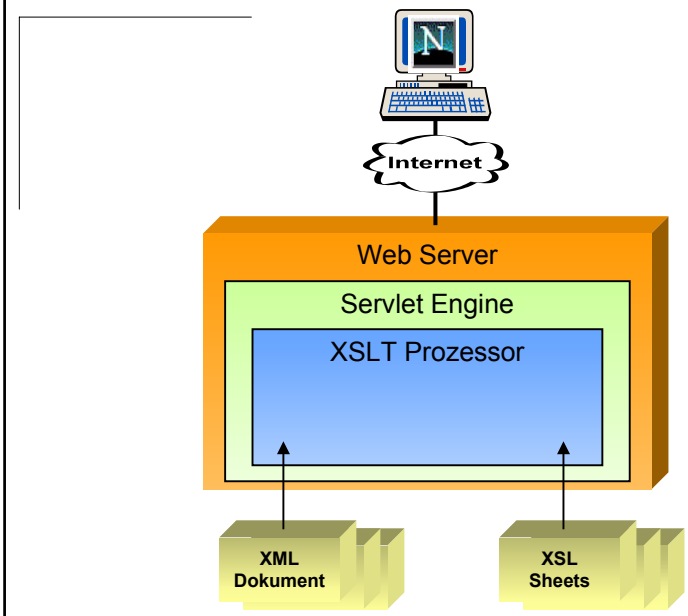
XSLT Prozessor

Orientation in Objects



© 2000-2001 Orientation in Objects GmbH

Idee: XSLT im Servlet



© 2000-2001 Orientation in Objects GmbH

- Apache Cocoon is a 100% pure Java publishing framework that relies on new W3C technologies (such as DOM, XML, and XSL) to provide web content.

Quelle: Cocoon 1 README

A small revolution

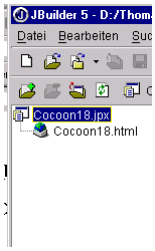
Stefano Mazzocchi (Autor und Architekt von Cocoon)

© 2000-2001 Orientation in Objects GmbH

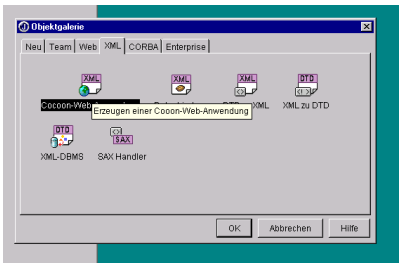
JBuilder5 und Cocoon

Orientation in Objects

Cocoon in ein Projekt einbinden :



Datei -> Neu

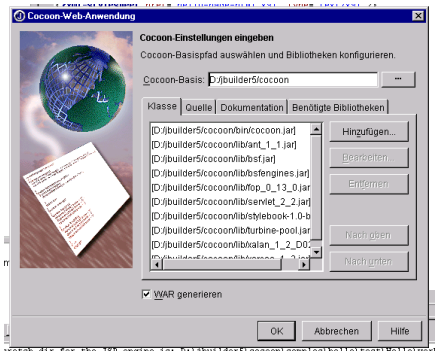


© 2000-2001 Orientation in Objects GmbH

JBuilder5 und Cocoon

Orientation in Objects

Cocoon in ein Projekt einbinden :



Konfiguration der Bibliotheken

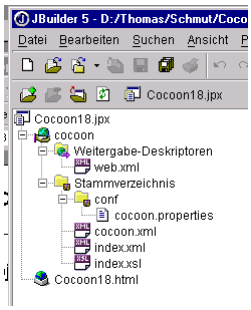
- Cocoon - Basis
- Klasse
- Quelle
- Dokumentation
- Bibliotheken
- WAR generieren

© 2000-2001 Orientation in Objects GmbH

JBuilder5 und Cocoon

Orientation in Objects

Projektfenster :



Dateien, die vom Cocoon - Experten erstellt werden

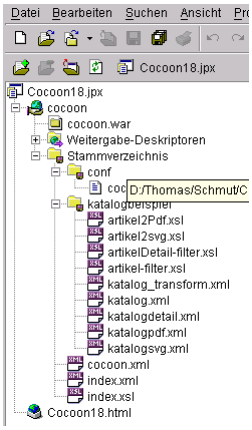
- Cocoon.war
- web.xml
- cocoon.properties
- cocoon.xml
- index.xml/xsl

© 2000-2001 Orientation in Objects GmbH

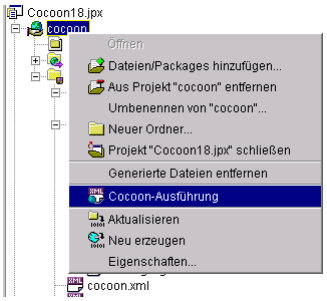
JBuilder5 und Cocoon

Orientation in Objects

Dateien in das Stammverzeichnis kopieren :



Cocoon ausführen :



© 2000-2001 Orientation in Objects GmbH

JBuilder5 und Cocoon

Orientation in Objects

Der Browser :

index

http://localhost:8080/cocoon/katalogbeispiel/katalog.xml

Pos.	ID	Bezeichnung
1	GTH 0015208	Injektionspflaster, 1,2cm * 4cm
2	GTH 0093131	Gärtnerbox
3	GTH 4051336	Heftpflaster standard, 1,25cm * 5m
4	GTH 4051342	Heftpflaster standard, 2,5cm * 5m
5	GTH 4051359	Heftpflaster standard, 5cm * 5m
6	GTH 4051365	Heftpflaster Vlies, 5cm * 10m
7	GTH 4051371	Heftpflaster Vlies, 2,5cm * 10m
8	GTH 4051684	Wundpflaster elastisch, 2cm * 6cm
9	GTH 4051690	Wundpflaster standard, 3cm * 6cm
10	GTH 4051738	Wundpflaster wasserabweisend, 2cm * 6cm

nächster

© 2000-2001 Orientation in Objects GmbH

Inhalt

Orientation in Objects

- XML - Konfiguration von JBuilder
- XML - Umwandlung und Validierung
- Web - Publishing mit Cocoon
- XML - Databinding mit Castor und BorlandXML
- XML - Datenbankkomponenten im JBuilder

© 2000-2001 Orientation in Objects GmbH

XML - Data - Binding

- Mit Hilfe von Datenbindung kann man auf Daten zugreifen, diese Bearbeiten und die bearbeiteten Daten an eine Datenbank zurückschicken oder mit einem XML – Dokument anzeigen.
- Das XML –Dokument fungiert als Übertragungsmechanismus zwischen Anwendung und Datenbank.

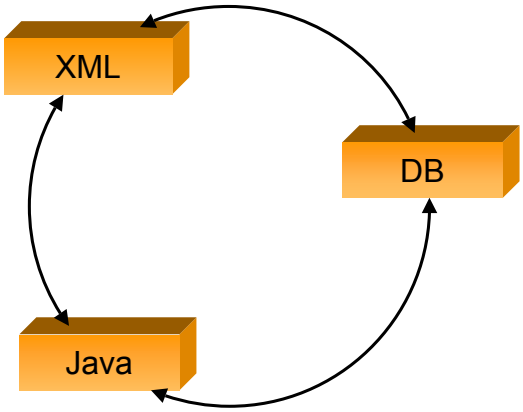
© 2000-2001 Orientation in Objects GmbH

XML - Data - Binding

- Die Datenbindung wird implementiert, indem Java Klassen generiert werden, die für die in einer Grammatik (DTD o. Schema) enthaltenen Begrenzungen stehen.
- Die XML – Files entsprechen dann den Instanzen der Klassen.
- Mit den so erstellten Klassen lassen sich XML – Dokumente lesen, erstellen oder validieren, die dieser Grammatik entsprechen.

© 2000-2001 Orientation in Objects GmbH

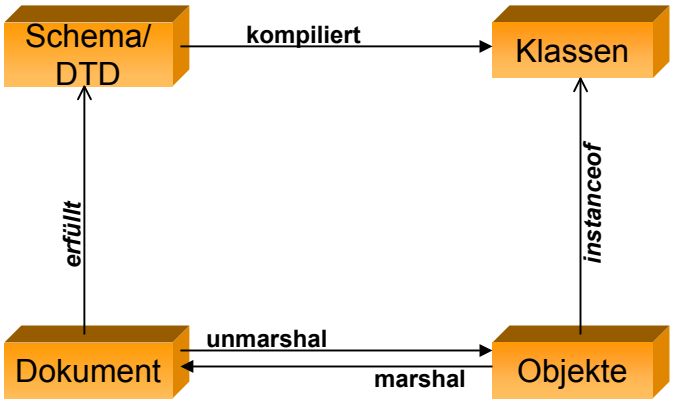
XML - Data - Binding



© 2000-2001 Orientation in Objects GmbH

XML - Data - Binding

„Roundtrip“



© 2000-2001 Orientation in Objects GmbH

Castor

Orientation in Objects

Das Objektmodell :

- Castor ist ein Datenbindungssystem, das eine Instanz eines XML – Schemas einem Objekt zuordnet das für die Daten steht.
- Das Objektmodell umfaßt Klassen, Typen sowie Deskriptoren, die verwendet werden, um Infos zu einer Klasse und deren Feldern abzurufen.

© 2000-2001 Orientation in Objects GmbH

Castor

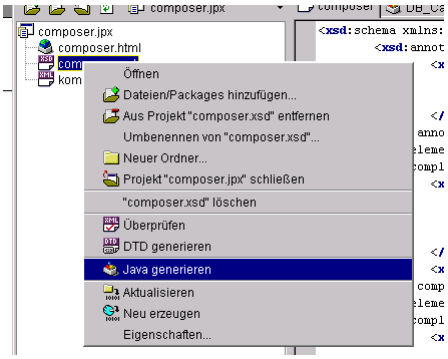
Orientation in Objects

Marshalling :

- Marshalling beschreibt einfach einen Vorgang, bei dem ein Objekt in einen Datenstrom umgewandelt wird und umgekehrt.
- Das Castor – System verwendet ein Marshalling – System, das ClassDescriptors und FieldDescriptors verwendet, um das Marhalling von und nach XML zu beschreiben.

© 2000-2001 Orientation in Objects GmbH

JBuilder5 und Castor

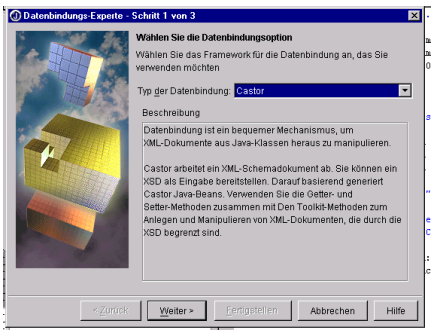


- Rechtsklick auf das Schema
- „Java generieren“ auswählen

© 2000-2001 Orientation in Objects GmbH

JBuilder5 und Castor

Der Datenbindungsexperte :



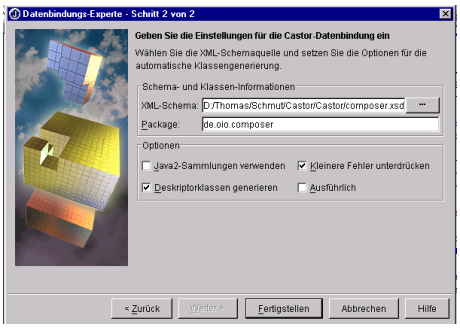
- Typ der Datenbindung „Castor“
- „Weiter“

© 2000-2001 Orientation in Objects GmbH

JBuilder5 und Castor

Orientation in Objects

Der Datenbindungsexperte für Castor:



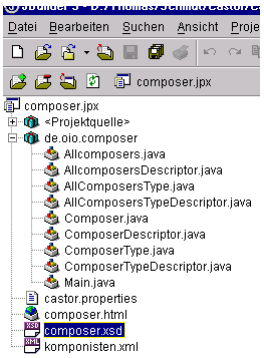
- Pfad des Schemas wird vorbesetzt
- Package - Namen angeben
- evtl. weitere Optionen festlegen
- „Fertigstellen“

© 2000-2001 Orientation in Objects GmbH

JBuilder5 und Castor

Orientation in Objects

Projektansicht :



- Java - Dateien, die vom Datenbindungsexperten erstellt wurden

© 2000-2001 Orientation in Objects GmbH

BorlandXML

Orientation in Objects

- BorlandXML verwendet DTDs um Java – Klassen zu generieren.
- Schritt 1 : Generierung einer Klassenmodelldatei aus einer DTD.
  Die Datei beschreibt eine übersichtliche Struktur der Zielklassen und ermöglicht es diese Klassen anzupassen
- Schritt 2 : BorlandXML generiert nun Java – Klassen aus der Klassenmodell – Datei .

© 2000-2001 Orientation in Objects GmbH

BorlandXML

Orientation in Objects

Der Datenbindungsexperte für BorlandXML:

© 2000-2001 Orientation in Objects GmbH

BorlandXML

Orientation in Objects

Funktionalität von BorlandXML

- JavaBean – Bearbeitung um ein XML – Dokument zu erstellen oder auf Daten im Dokument zuzugreifen.
- Marshalling und Unmarshalling .
- Dokumentenvalidierung : Überprüfen von JavaBean – Objekten vor dem Marshalling zu XML bzw. nach dem Unmarshalling eines Dokumentes zu JavaBean – Objekten .

© 2000-2001 Orientation in Objects GmbH

Inhalt

Orientation in Objects

- **XML - Konfiguration von JBuilder**
- **XML - Umwandlung und Validierung**
- **Web - Publishing mit Cocoon**
- **XML - Databinding mit Castor und BorlandXML**
- **XML - Datenbankkomponenten im JBuilder**

© 2000-2001 Orientation in Objects GmbH

XML - Datenbankkomponenten

Orientation in Objects

Typen von XML Dokumenten

- Datenzentrische XML - Dokumente
- Dokumentenzentrische XML -Dokumente

© 2000-2001 Orientation in Objects GmbH

XML - Datenbankkomponenten

Orientation in Objects

datenzentrische XML - Dokumente

- XML dient als Transportmedium zwischen DB und Anwendung
- Daten werden zur maschinellen Weiterverarbeitung verwendet
- sehr hohe Granularität der Daten
- regelmäßige Struktur
- keine Mischinhalte

© 2000-2001 Orientation in Objects GmbH

XML - Datenbankkomponenten

datenzentrisches XML - Dokument

```
<SalesOrder SONumber="12345">
    <Customer CustNumber="543">
      <CustName>ABC Industries</CustName>
      <Street>123 Main St.</Street>
      <City>Chicago</City>
      <State>IL</State>
      <PostCode>60609</PostCode>
    </Customer>
....
</SalesOrder>
```


© 2000-2001 Orientation in Objects GmbH

XML - Datenbankkomponenten

dokumentenzentrische XML - Dokumente

- eher unregelmäßige Struktur
- weniger granuliert
- viel Mischinhalt
- die Reihenfolge des Auftauchens der PCDATA und Childelemente ist wichtig

© 2000-2001 Orientation in Objects GmbH

XML - Datenbankkomponenten

Orientation in Objects

dokumentenzentrisches XML - Dokument

```
<Produkt>
   <Name>Bananenquark</Name>
   <Hersteller>Frutti GmbH Hinterdupfingen</Hersteller>
   <Kurzbeschreibung>leckere Bananenquark aus dem Hinterland</Kurzbeschreibung>
   <Beschreibung>
      <Para>Der leckere Bananenquark wird in <i>farbigen Plastikschalen</i>, die <b>
            völlig ökologisch</b> gereinigt wurden, ausgeliefert.
            Die verwendeten Bananen wurden aus den <b>
            besten Banenen Südschwedens</b> ausgewählt. ....
      </Para>
      <Para>Sie können:</Para>
      <Liste>
      <Eintrag>
         <Link dest="bestellen.html">gleich Bananenquark bestellen </Link>
      </Eintrag>
      <Eintrag>
         <Link dest="schweden.html">mehr über den Auswahlprozess der Bananen
                                    in Schweden erfahren </Link>
      </Eintrag>
            .....
</Produkt>
```


© 2000-2001 Orientation in Objects GmbH

Dokumente in die Datenbank bringen

Orientation in Objects

Software, die Daten zwischen Dokumenten und relationalen Datenbanken austauscht, benötigt ein sogenanntes „Mapping“. Hierbei unterscheidet man zwischen zwei Formen:

- Vorlagenbasiertes Mapping (Template - Driven Mapping)
- Modellbasiertes Mapping (Model - Driven Mapping)

© 2000-2001 Orientation in Objects GmbH

Vorlagenbasiertes Mapping

Orientation in Objects

- Es gibt kein vordefiniertes mapping zwischen Dokument und Datenbank
- produktspezifische Kommandos werden in eine Vorlage eingebettet, die von der Übertragungssoftware ausgeführt werden
- Die Middleware ersetzt die Anweisung durch die korrespondierenden Daten und formatiert das Ganze als XML

© 2000-2001 Orientation in Objects GmbH

Vorlagenbasiertes Mapping

Orientation in Objects

Vorlage :

```
<FlightInfo>
        <Introduction>The following flights have availablseats:</Introduction>
        <SelectStmt>SELECT Airline,        FltNumber, Depart, Arrive FROM Flights</SelectStmt>
        <Conclusion>We hope one of these meets your needs</Conclusion>
 </FlightInfo>
```

Rückgabe :

```
<FlightInfo>
        <Introduction>The following flights have available seats:</Introduction>
        <Flights>
                <Row>
                        <Airline>ACME</Airline>
                        <FltNumber>123</FltNumber>
                        <Depart>Dec 12, 1998 13:43</Depart>
                        <Arrive>Dec 13, 1998 01:21</Arrive>
                </Row> ...
        </Flights>
        <Conclusion>We hope one of these meets your needs.</Conclusion>
</FlightInfo>
```


© 2000-2001 Orientation in Objects GmbH

Modellbasiertes Mapping

Orientation in Objects

- Daten entsprechen einer vordefinierten Form und werden dementsprechend „gemapped“
- Gegenüber dem template-driven Mapping wird Flexibilität eingebüßt, allerdings wird die Einfachheit erhöht.
- Dokumente, die nicht der vordefinierten Form entsprechen, können per XSLT transformiert werden.

© 2000-2001 Orientation in Objects GmbH

Modellbasiertes Mapping

Orientation in Objects

Map - File :

```
<XMLToDBMS Version="1.0">
  <Options>
  </Options>
  <Maps>
    <ClassMap>
      <ElementType Name="composer"/>
      <ToClassTable>
      <Table Name="composer"/>
      </ToClassTable>
      <PropertyMap>
        <Attribute Name="epoche"/>
        <ToColumn>
          <Column Name="epoche"/>
        </ToColumn>
      </PropertyMap>
      <PropertyMap>
        <ElementType Name="name"/>
        <ToColumn>
          <Column Name="name"/>
        </ToColumn>
   </ClassMap>
  </Maps>
</XMLToDBMS>
```


© 2000-2001 Orientation in Objects GmbH

Tabellen und Objektrelationales Modell

Orientation in Objects

Sichtweisen der Abbildung von Daten in einem XML - Dokument :

- als Tabellenmodell und daraus resultierend das tabellenbasierte Mapping
- als Objektmodell und daraus resultierend das objekt – relationale Mapping

© 2000-2001 Orientation in Objects GmbH

Tabellenbasiertes Mapping

Orientation in Objects

- Wird oft von Middleware Produkten benutzt, um Daten zwischen XML Dokumenten und relationalen Datenbanken auszutauschen.
- XML Dokumente werden als einzelne oder mehrere Tabellen in der Datenbank modelliert.
- Die XML - Dokumente müßen einer bestimmten Struktur genügen

© 2000-2001 Orientation in Objects GmbH

Tabellenbasiertes Mapping

Struktur eines XML - Files und die entsprechende Tabelle:

```
<A>
   <B>
      <C>ccc</C>                Table A
      <D>ddd</D>                -------
      <E>eee</E>             C     D     E
   </B>                     ---   ---   ---
   <B>                <=>   ...   ...   ...
      <C>fff</C>            ccc   ddd   eee
      <D>ggg</D>            fff   ggg   hhh
      <E>hhh</E>            ...   ...   ...
   </B>
</A>
```


© 2000-2001 Orientation in Objects GmbH

Objekt - Relationales Mapping

- Wird von der XML -DBMS Middleware im JBuilder verwendet
- Beim Objekt - Relationalen Mapping werden die Daten des XML Dokumentes als Baum von Objekten modelliert

© 2000-2001 Orientation in Objects GmbH

Objekt - Relationales Mapping

XML - File, Objektmodell und Tabellenmodell

```
<SalesOrder>
      <Number>1234</Number>
      <Customer>Gallagher Industries</Customer>
      <Date>29.10.00</Date>
      <Item Number="1">
         <Part>A-10</Part>
         <Quantity>12</Quantity>
         <Price>10.95</Price>
      </Item>
      <Item Number="2">
         <Part>B-43</Part>
         <Quantity>600</Quantity>
         <Price>3.99</Price>
      </Item>
   </SalesOrder>
```

```
object SalesOrder {
        number = 1234;
        customer = "Gallagher Industries";
        date = 29.10.00;
        items = {ptrs to Item objects};
     }
                 /    \
                /      \
               /        \
   object Item {        object Item {
      number = 1;          number = 2;
      part = "A-10";       part = "B-43";
      quantity = 12;       quantity = 600;
      price = 10.95;       price = 3.95;
   }                    }
```

```
SaleOrders
   ----------
   Number   Customer                 Date
   ------   --------------------     --------
   1234     Gallagher Industries     29.10.00
   ...      ...                      ...

Items
   -----
   SONumber   Item   Part   Quantity   Price
   --------   ----   ----   --------   -----
   1234       1      A-10   12         10.95
   1234       2      B-43   600        3.99
   ...        ...    ...    ...        ...
```


© 2000-2001 Orientation in Objects GmbH

Vorlagenbasierte Komponeneten in JBuilder

JBuilder kommt mit zwei vorlagenbasierten Komponenten, XTable und XQuery.
Die Komponenten sind Unidirektional, es können nur Daten aus der Datenbank herausgezogen werden .
Die Parametrisierung dieser Komponenten kann auf drei Arten erfolgen :

- über den Customizer der Komponente
- über den Inspektor
- mit Hilfe eines XML - Abfragedokumentes

© 2000-2001 Orientation in Objects GmbH

Vorlagenbasierte Komponeneten in JBuilder

XTable :

- führt eine Select-Anweisung auf eine Untermenge einer Tabelle aus.
- Die Untermenge ergibt sich aus der Bedingung (Schlüssel-Wert-Paare), die in der where-Klausel der Select-Anweisung definiert ist.
- Bei der Ausführung wird eine Ergebnismenge in XML generiert.
- Der Tabellenname, Schlüssel-Wert-Paare und andere Parameter werden entweder in einem XML-Abfragedokument, in Bean-Eigenschaften oder in beidem bereitgestellt

© 2000-2001 Orientation in Objects GmbH

Vorlagenbasierte Komponeneten in JBuilder

XQuery :

- Führt eine SQL-Anweisung aus und generiert eine Ergebnismenge in XML.
- SQL und weitere Parameter werden entweder in einem XML-Abfragedokument, in Bean-Eigenschaften oder in beidem bereitgestellt.
- der Unterschied zu XTable besteht darin, dass die Komponente direkt eine SQL - Anweisung entgegennimmt.

© 2000-2001 Orientation in Objects GmbH

Modellbasierte Komponenten in JBuilder

JBuilder implementiert dazu die Open - Source Middleware XML-DBMS.
Diese Lösung benötigt folgende Komponenten:

- relationale Datenbank mit einem JDBC - Treiber
- XML - Dokument zur ein und Ausgabe von Daten
- Map - File
- API - Methoden zu Datenübertragung von und zur Datenbank

© 2000-2001 Orientation in Objects GmbH

Modellbasierte Komponenten in JBuilder

JBuilder bietet folgende XML - DBMS - Unterstützung :

- XML - DBMS - Experte
- die Komponente XMLDBMSTable
- die Komponente XMLDBMSQuery

© 2000-2001 Orientation in Objects GmbH

Modellbasierte Komponenten in JBuilder

Der XML - DBMS Experte :

- Erstellen einer SQL - Script - Datei auf Grundlage einer DTD
- Erstellen einer Map - Datei
- noch keine Untersützung zu Erstellung von Map - Dateien aus Datenbankschemata

© 2000-2001 Orientation in Objects GmbH

Modellbasierte Komponenten in JBuilder

XMLDBMSQuery und XMLDBMSTable :

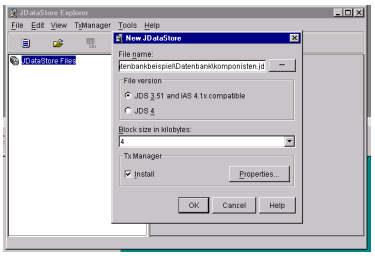
- Diese Beans werden verwendet, um Datenbankdaten in ein XML-Dokument bzw. XML - Dokumente in Datenbankdaten umzuwandeln.
- Sie werden in Verbindung mit der MAP-Datei und Datenbanktabellen verwendet, die durch den Experten zum Generieren von XML-DBMS-Mapping und SQL erzeugt wurden.
- der Unterschied zwischen den Komponenten besteht darin, dass die XMLDBMSQuery - Komponente eine SQL - Anweisung entgegennimmt und nur Unidirektional ist.

© 2000-2001 Orientation in Objects GmbH

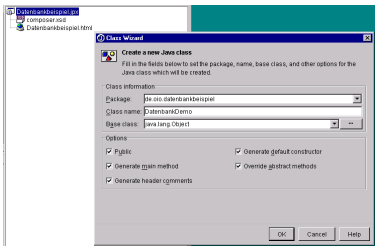
Beispiel zu XMLDBMSQuery und XMLDBMSTable

Orientation in Objects

Neue Datenbank anlegen :



Klasse zum Projekt hinzufügen :

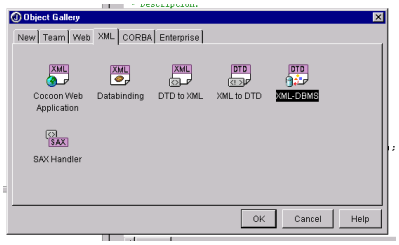


© 2000-2001 Orientation in Objects GmbH

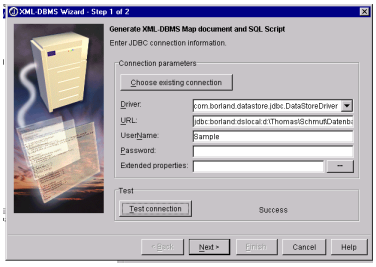
Beispiel zu XMLDBMSQuery und XMLDBMSTable

Orientation in Objects

XML DBMS - Experten aufrufen :



JDBC - Einstellungen :

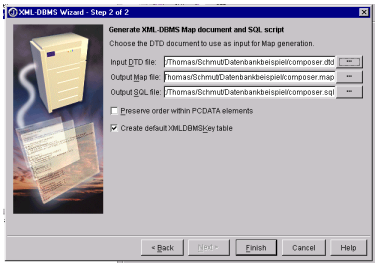


© 2000-2001 Orientation in Objects GmbH

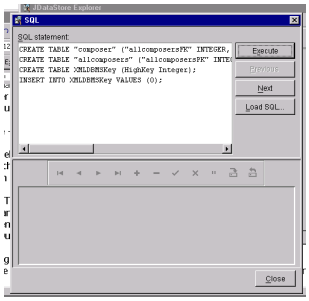
Beispiel zu XMLDBMSQuery und XMLDBMSTable

Orientation in Objects

Map und SQL - File erzeugen :



Tabellen erzeugen :

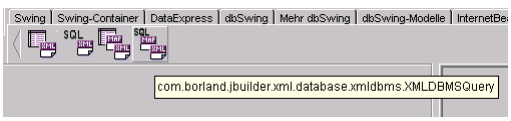


© 2000-2001 Orientation in Objects GmbH

Beispiel zu XMLDBMSQuery und XMLDBMSTable

Orientation in Objects

XMLTable - Bean über den Designer hinzufügen :

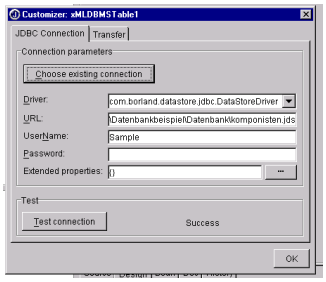


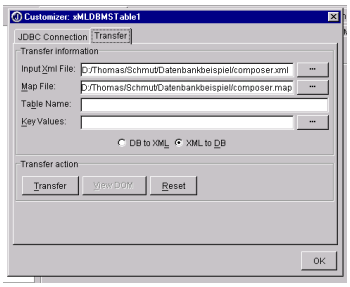
© 2000-2001 Orientation in Objects GmbH

Beispiel zu XMLDBMSQuery und XMLDBMSTable

Orientation in Objects

XMLTable - Bean über den Customizer konfigurieren :



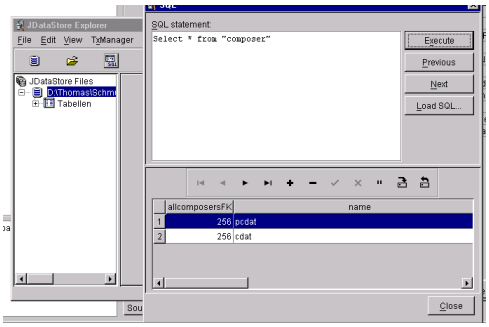


© 2000-2001 Orientation in Objects GmbH

Beispiel zu XMLDBMSQuery und XMLDBMSTable

Orientation in Objects

Überprüfen, ob die Übertragung funktioniert hat :



© 2000-2001 Orientation in Objects GmbH

Fragen

Orientation in Objects

© 2000-2001 Orientation in Objects GmbH

Quellen

Orientation in Objects

Bookmarks :

Seybold Publications and O'Reilly & Associates,Inc :

- www.xml.com

Dokumentation zu XML - DBMS

- www.rpbourret.com

JBuilder5 :

- www.borland.com
- JBuilder5 Online - Hilfe

Cocoon Home :

- xml.apache.org

Castor Home :

- castor.exolab.org

© 2000-2001 Orientation in Objects GmbH

Ende

Orientation in Objects

Danke für Ihre Aufmerksamkeit!

© 2000-2001 Orientation in Objects GmbH