



Infonyte GmbH
Julius-Reiber-Str. 15
D-64293 Darmstadt
0700 INFONYTE
www.infonyte.de



Skalierbare XML- Verarbeitung mit Infonyte

Dr. Thomas Tesch
Geschäftsführer
<<mailto:tesch@infonyte.de>>

Leipzig, 27. Februar 2003

Einsatzschwerpunkte XML



<http://www.infonyte.de>

Wo wird XML heute erfolgreich eingesetzt?

XML als Integrations-Glue

- Business Process Integration, Enterprise Application Integration
- Kopplung von Mainframe, Legacy-Anwendungen, ERP-Software und Web-Anwendungen
- Bedarfsgerechte Datenintegration

Medienneutrales Publizieren

- Informationen bedarfsgerecht für bestimmte Zielmedien zusammensetzen
- Verwaltung, Durchsuchbarkeit grosser Dokumentenbestände
- Authoring, Production, Deployment

we scale your XML

Datenintegration

infonyte

<http://www.infonyte.de>

we scale your XML

XML-Vorteile bei der Kopplung von Systemen

- Ausdrucksmächtigkeit: verlustfreie Darstellung heterogener Datenformate (EDI, Relational, etc.)
- Flexibilität: Kombination von Daten und Metadaten
- Plattformunabhängigkeit: textuelles Format, Unicode Unterstützung

Ansatz

- inkrementelle, bedarfsgerechte Datenintegration
- Entkopplung der Systeme durch XML-Warehouse
- Verarbeitung mit XML-Standards

© Infonyte GmbH

Medienneutrales Publizieren

infonyte

<http://www.infonyte.de>

we scale your XML

XML-Vorteile in medienneutralen Dokumentations- und Publikationsprozessen

- Qualitätssicherung: Validierungswerkzeuge ermöglichen die Durchsetzung von einheitlichen Dokumentationsstandards.
- Austauschbarkeit: Internationale Dokumentenstandards (z.B. AECMA , DocBook)
- Wiederverwendbarkeit: Trennung von logischer Struktur und Layout ermöglicht die Bedienung unterschiedlicher Medien (Handbuch, CD-ROM, Web) und Benutzergruppen aus einer Quelle (leichte Wartbarkeit).
- Herstellerunabhängigkeit durch Verwendung offener Standards

© Infonyte GmbH

Unterschiedliche Anforderungen

infonyte

<http://www.infonyte.de>

Je nach Anwendungsfall unterschiedliche Anforderungen

we scale your XML

XML-Nutzung

- Schemanutzung: wohlgeformtes XML, DTD/Schema
- Struktur: datenorientiert, dokumentenorientiert
- Reihenfolge: datenorientiert, dokumentenorientiert

XML-Verarbeitung

- Herkunft / Erzeugung von XML-Daten
- Zugriffsgranularität
- Lesender/Schreibender Zugriff
- Umgang mit Einzeldokumenten oder Dokumentkollektionen
- Inhaltsorientierter oder strukturorientierter Zugriff

© Infonyte GmbH

Herausforderungen

infonyte

<http://www.infonyte.de>

XML Verarbeitung heute

- Integration in IT-Architekturen
- Komplexität der Anwendungsprogrammierung
- XML-Ströme, temporäre Speicherung von XML-Daten

Ressourcenhungrige Verarbeitungsprozesse

- Verbosität: textuelles XML ist speicher- und bandbreitenhungrig
- Komplexe Standards: DOM-API, XSLT-Verarbeitung

Neue Werkzeuge zur skalierbaren XML-Verarbeitung

- Speicherung von XML ...
- ... zur effizienten Verarbeitung von XML
- komplementär zu bestehenden relationalen Systemen

we scale your XML

© Infonyte GmbH

XML-Verarbeitung (1)

Datei-orientierte Verarbeitung

- Einlesen (SAX)
- Interne Repräsentation (DOM, nativ)
- Re-Serialisierung Speicherung
- Speicherverbrauch (5-20 Byte pro XML-Byte)
- Overhead durch Serialisierung
- zur Verarbeitung grosser Dokumente ungeeignet
- keine weiteren Dienste (Index, Kollektion, Integrität)
- + für die Verarbeitung kleiner XML-Dokumente
- + Zugriff über die XML-Standards möglich (DOM, XPath, XSLT)
- + freie Tools verfügbar (Apache)

XML-Verarbeitung (2)

Verarbeitung mit RDBMS, OODBMS, OR-Mappers, ...

- Schemadefinition
- Abbildung in physisches Datenmodell
- Aufwendige Infrastruktur
- Abbildungen unterstützen nicht alle XML-Features (PI, Comments)
- Zugriff über XML-Standards (XPath, XSLT, DOM) nicht möglich
- Impedance Mismatch: XML-Sicht nicht update-fähig
- oftmals redundante Speicherung als BLOB
- schlechte RDBMS Performanz für dokumentorientiertes XML
- Serialisierung der XML-Dokumente aufwendig
- + für die Generierung von datenorientierten XML aus relationalen Daten

Infonyte-DB

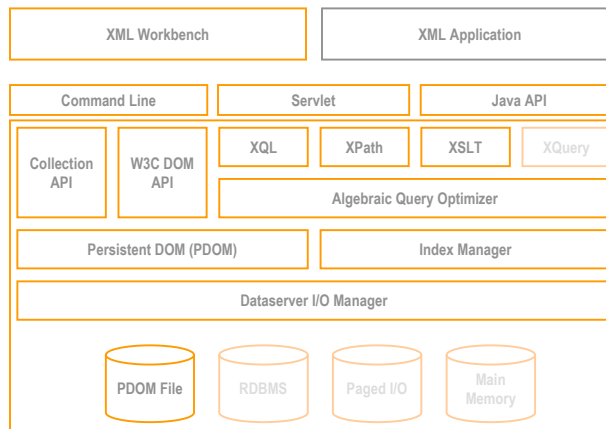
Ziele

- schlanker und skalierbarer XML-Kern
- Konzentration auf die anwendungsübergreifende Funktionalität
- Kommunikation über standardisierte Schnittstellen

Infonyte-DB

- Natives XML Datenmodell
 - PDOM: kompaktes, indiziertes, binäres XML-Format
 - verarbeitet direkt wohlgeformtes XML
- Persistent XSLT Processing (PXSLT)
- Self contained: keine weiteren Backend Systeme
- Modularer Aufbau, leichte Integrierbarkeit
- Standards: W3C DOM, XPath, XSLT
- Code-Grösse 500-800KB, ab 16MB Hauptspeicher

Architektur



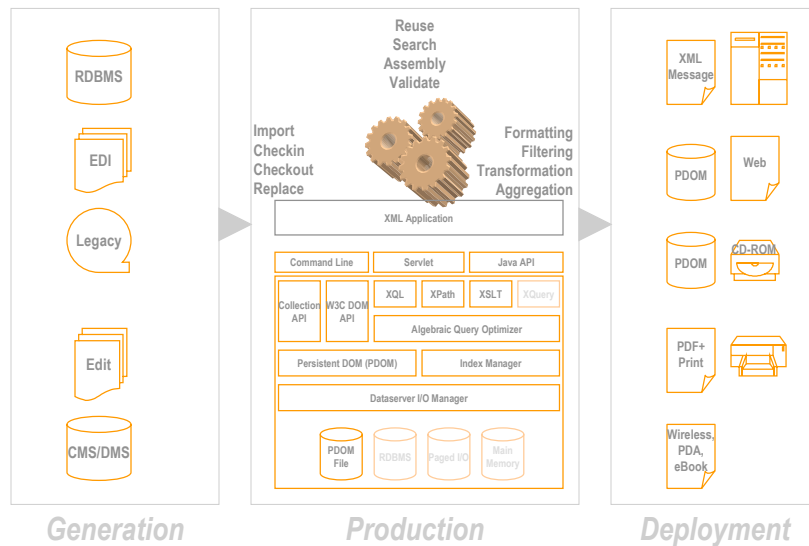
Produkt und Kundennutzen

we scale your XML

- Native XML Speicherung
 - Well-formed XML
- Einfache Abfrage
 - XQL, XPath, XSLT
 - Integration mit Apache Xalan
- Skalierbarkeit
 - PDOM: transp. Hauptspeicherzugriff
 - Selbstoptimierender Cache
 - Speichert bis zu 2³⁰ XML Knoten
 - Indexmanagement
- Small footprint
 - 30%-100% Platzersparnis
 - Hauptspeicher 800K Minimum
- Standards
 - W3C DOM Core Level 2 API
 - W3C XPath 1.0 + XSLT 1.0
 - Namespaces, TrAX, JAXP
- 100% Java

Verarbeitung

we scale your XML



PXSLT

we scale your XML

Skalierbare XSLT Verarbeitung

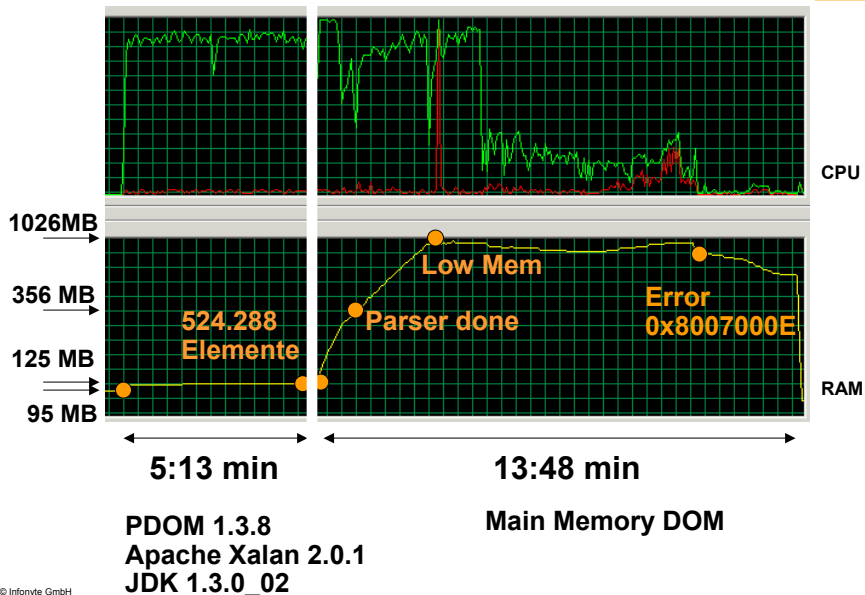
- Hauptspeicheranforderungen der XSLT-Prozessoren
- Alternative: forward-only traversal durch SAX-Verarbeitung
- Alternative: Segmentierung der zu verarbeitenden Daten

Infonyte PXSLT

- arbeitet direkt auf dem PDOM Format
- PDOM-Erzeugung in linearer Zeit
- Mehrfachtransformationen ohne erneute Serialisierung möglich
- Unterstützung für XPath-Ausdrücke and Match-Patterns durch Struktur Index
- Konfigurierbarer Cache
- konstanter Hauptspeicherbedarf

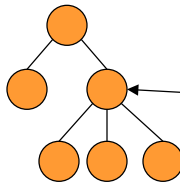
XSLT Verarbeitung

we scale your XML



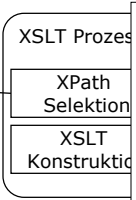
Layout: XSLT, XSL-FO

we scale your XML



XML
Quelldoku-
ment(e)

```
<weather-information>  
<location>  
<city>Darmstadt</city>  
</location>  
<summary>  
Sonnig,  
wolkenlos, 20 C,  
nachts 8 C  
</summary>  
</weather-information>
```

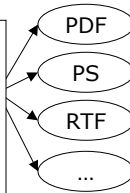


```
<xsl:stylesheet>  
<xsl:template match="/">  
... presentation info here ...  
</xsl:template>  
  
<xsl:template  
match="weather-information[location/city =  
'Darmstadt']">  
<xsl:apply-templates select="summary"/>  
...  
</xsl:template>  
</xsl:stylesheet>
```

XSLT/XPath
Stylesheet

```
<?xml version="1.0"?>  
<fo:root  
xmlns:fo="http://www.w3.org/XSL/Format/1.0">  
...  
<fo:flow font-size="14pt" line-height="14pt">  
<fo:block text-align="centered"  
font-size="24pt" line-height="28pt">  
Wettervorhersage Darmstadt  
</fo:block>  
...  
</fo:flow>  
</fo:root>
```

Zieldokument



PDOM

we scale your XML

Skalierbare Speicherung

- XML-Speicherung in RDBMS ist aufwendig, Impedance Mismatch
- DOM Realisierungen sind i.d.R. Hauptspeicherintensiv

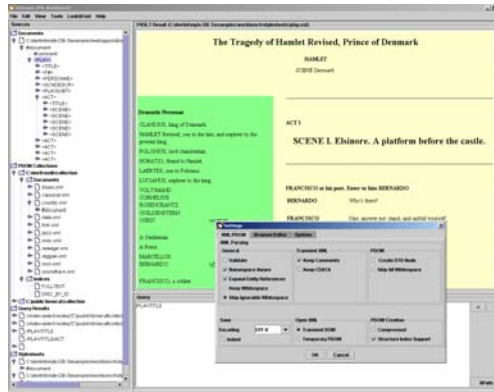
Infonyte PDOM

- persistente W3C-DOM Level 2 Implementierung
- unterstützt Dokumentensammlungen
- kompaktes, verlustfreies binäres Speicherformat
- Größe der XML Daten kann um bis zu 70% reduziert werden
- Strukturindex, Datenindex, Volltextindex
- Transaktionskonzepte: Atomarität und Dauerhaftigkeit
- konfigurierbarer Cache
- konstante Hauptspeicheranforderungen

XML-Workbench

InfonYTE XML-Workbench

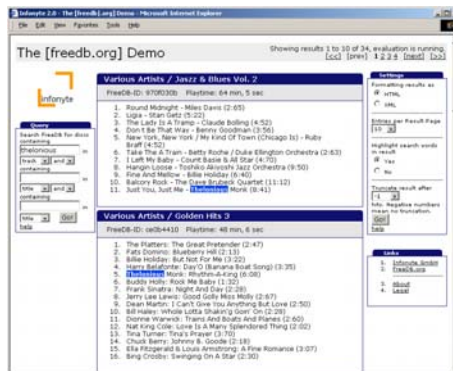
- grafische Arbeitsumgebung für XML Dokumente, Kollektionen, Stylesheets
- voll integriert mit InfonYTE-DB
- Erstellung, Änderung, Abfrage, Transformation und Formatierung von Dokumenten
- Integration von HTML und PDF Werkzeugen
- Konfiguration der InfonYTE-DB Settings



Performanz

FreeDB, <http://www.freedb.org>

- 500.000 CD-Beschreibungen, ~500MB als XML-Rohtext
- Standard-PC (1,8 Ghz, 512 MB RAM)
- Parsen: 32 Millionen XML-Knoten, ~4 Minuten, 400MB
- Index CD-Schlüssel: 548.000 Knoten, 1,7%, 88 Sekunden
- Volltext-Index: 28 Millionen Knoten, 89%, 17 Minuten, 90MB
- XSLT Verarbeitung: Durchsatz 10-20MB/Sekunde



Anwendungsszenario (1)

Szenario

- Web-Publishing von Finanz- und Handelsdaten
- Daten stammen aus Mainframe

Herausforderungen

- bis zu 10GB Datenvolumen pro Tag
- Entkopplung vom Geschäftsprozess
- dynamische Transformation auf einem Web-Server
- Hoher Durchsatz bei XSLT-Verarbeitung

Lösung

- Anbindung Infonyte-DB an Mainframe zum XML-Import
- Bereitstellung effizienter Indexstrukturen
- Integration in Web-Server Architektur

Anwendungsszenario (2)

Szenario

- IETD: Interaktive Elektronische Technische Dokumentation
- Technische Dokumentation im Luftfahrtbereich
- Well-formed XML, aus SGML

Herausforderungen

- Formatierung elektronischer Handbücher, verschiedene Zielgruppen
- leistungsfähige Lese- und Navigationswerkzeuge (Standard-SW)
- kostengünstige und sichere Verteilung
- (feingranulare Updates)

Lösung

- Infonyte-DB als Produktions- und Deployment Lösung
- PDOM Dateien als Distributionsformat
- zusätzliche Integration in Web-Server Architektur

Anwendungsszenario (3)

Szenario

- Interaktive Dokumentation/Zugriff auf eBooks
- Lokale Verarbeitung auf Mobilgeräten (PDA, PocketPC)

Herausforderungen

- Begrenzte Systemressourcen
- Personalisierung
- Synchronisation mit Server



Lösung

- PDOM als skalierbarer und intelligenter Cache
- Nutzung zusätzlicher Eigenschaften wie XML-Kompression
- XPath als kompakte Repräsentation von Sichten

Zusammenfassung

XML-Anwendungen unterscheiden sich

- unterschiedliche Nutzung von XML-Merkmalen
- unterschiedliche Anforderungen an die XML-Verarbeitung

Skalierbarer XML-Kern als Motor für flexible Datenintegrationsprozesse

- Klein, schnell, erweiterbar, ...
- Leichte Integrierbarkeit, 100% Java, 100% XML
- Konform zu den W3C-Standards

Fragen?

infonyte

<http://www.infonyte.de>

we scale your XML

```
<?xml version="1.0"?>
<talk id="BTW, Leibzig">
  <speaker>
    <person> Thomas Tesch </person>
  </speaker>
  <thanks>
    <person> Peter Fankhauser </person>
    <person> Gerald Huck </person>
    <person> Ingo Macherius </person>
  </thanks>
  <contact>
    http://www.infonyte.de
  </contact>
</talk>
```