DB2 UDB Extenders: XML & Text Extender

DDUG 20 November 2002

Pieter Bedert

ABIS Training & Consulting
Postbus 122, NL-3440 AC Woerden, tel: +31-348-435570
Postbus 220, BE-3000 Leuven, tel: +32-16-245610

http://www.abis.be - e-mail: training@abis.be

Agenda

Introducing Extenders

Text Extender

- Features
- DB2 UDB Implementation
- Enabling the Text Extender / Demo

XML Extender

- Introducing XML
- Features
- DB2 UDB Implementation
- Enabling the XML Extender / Demo

Text Extender for XML documents

- Enabling / Demo

Introducing Extenders

What is a DB2 Extender?

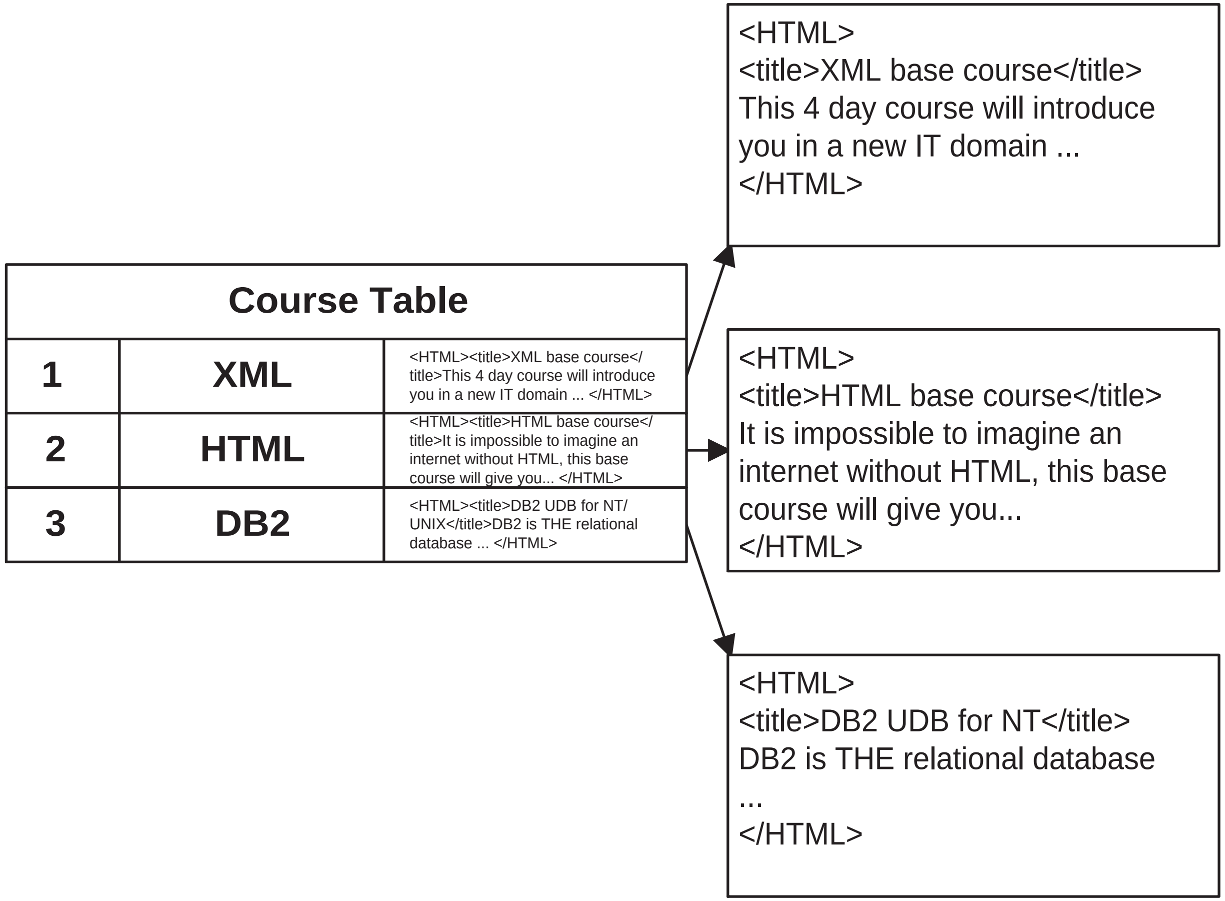
An Extender offers you

- Stored Procedures
- Triggers
- User Defined Functions (UDF)
- User Defined Datatypes (UDT)
- Supporting Tables

to Extend the DB2 functionality!

Text Extender: Features

Search for course in “HTML”?



Course Table		
1	XML	<HTML><title>XML base course</title>This 4 day course will introduce you in a new IT domain ... </HTML>
2	HTML	<HTML><title>HTML base course</title>It is impossible to imagine an internet without HTML, this base course will give you... </HTML>
3	DB2	<HTML><title>DB2 UDB for NT/UNIX</title>DB2 is THE relational database ... </HTML>

```
<HTML>
<title>XML base course</title>
This 4 day course will introduce
you in a new IT domain ...
</HTML>
```

```
<HTML>
<title>HTML base course</title>
It is impossible to imagine an
internet without HTML, this base
course will give you...
</HTML>
```

```
<HTML>
<title>DB2 UDB for NT</title>
DB2 is THE relational database
...
</HTML>
```

Text Extender: Features

When enabling Text for search by DB2,

DB2 will:

- Ignore “stop words”
  - the, and, or,...
  - list can be modified by the DBA / application developer
- Ignore tags
  - XML-tags, HTML-tags,...
  - kind of text must be indicated by DBA / application developer

DB2 UDB Extenders: XML & Text Extender

Text Extender: Features

When enabling Text for search by DB2,

DB2 users will be able to:

- Search documents using SQL with UDF’s

```
SELECT course_id
FROM course_table
WHERE db2tx.contains(coursedesc_texth, ‘”HTML”’) = 1


SELECT course_id, db2tx.no_of_matches(coursedesc_texth, ‘”HTML”’) as #occurences
FROM course_table
WHERE db2tx.contains(coursedesc_texth, ‘”HTML”’) = 1
ORDER BY 2 DESC
```

Text Extender: DB2 UDB Implementation

When you enable

- a Column (varchar / CLOB)
- an External File

for Text Search in a DB2 table,

DB2 will:

- Build an INDEX
- Add a HANDLE-COLUMN.

Text Extender: DB2 UDB Implementation

Index Options = Search Options

- **PRECISE Index**
  - “exact word” index
  - “stop words” ignored
- **LINGUISTIC Index**
  - “stemmed word” index
  - “stop words” ignored
  - dictionary needed! (Dutch is available)
- **DUAL Index**
  - PRECISE index + LINGUISTIC index
- **NGRAM Index**
  - all words entered “as is”
  - character based search
  - fuzzy search support

Text Extender: DB2 UDB Implementation

Consider: After

- INSERT
- UPDATE
- DELETE

NO AUTOMATIC INDEX SYNCHRONIZATION!

The DBA / Application Developer has to

- manually update the index
- schedule the automatic index updates

Text Extender: Enabling the Text Extender / Demo

Creating and Starting a Text Extender service (NT) / deamon (UNIX)

Create Text Extender Instance:

> txicrt

z/OS and OS/390: Use the DESCUST Utility in the USS

Drop Text Extender Instance:

> txidrop

Start Text Extender Instance:

> txstart

Stop Text Extender Instance:

> txstop

Text Extender: Enabling the Text Extender / Demo

Using the Text Extender Administration Interface,

```
> db2tx ...
```

Prepare (enable) your database for Text Extender use.

```
> db2tx enable database
```

z/OS and OS/390 >> db2tx ENABLE SERVER FOR DB2TEXT

Configure the Default Index values.

```
> db2tx change text cfg using ...
> db2tx get text configuration
```

Prepare (enable) your table / column for Text Extender use.

```
> db2tx enable text column “table_name” “column_name” handle “handle_column”
```

Maintain (update/check status) the index

```
> db2tx update index “table_name” handle “handle_column”
> db2tx get index status “table_name” handle “handle_column”
```

Text Extender: Enabling the Text Extender / Demo

From now on SQL will do the job:

> db2 insert into ...

> db2 update ...

> db2 delete ...

Don’t forget to update the index!

> db2tx update index ...

DB2 UDB Extenders: XML & Text Extender

Text Extender: Enabling the Text Extender / Demo

Searching the text:

```
SELECT ...
FROM ...
WHERE ...
```

Using the Text Extender UDF’s:

```
SELECT course_id
FROM course_table
WHERE db2tx.contains(coursedesc_texth, ‘”HTML”’) = 1
```

```
SELECT course_id, db2tx.no_of_matches(coursedesc_texth, ‘”HTML”’) as #occurences
FROM course_table
WHERE db2tx.contains(coursedesc_texth, ‘”HTML”’) = 1
ORDER BY 2 DESC
```

XML Extender: Introducing XML

What is an XML-document?

- Adds metadata to your data
- Hierarchical structure
- Consists of Elements and Attributes

```
<Course id=”1”>
    <Title>DB2 UDB for OS/390<\Title>
    <Description>This Course reveals the secrets of the leading RDBMS ... </Description>
    <Prereq_list>
        <Prereq>Know the concepts of a relational DataBase</Prereq>
        <Prereq>Know some basic SQL</Prereq>
        <Prereq>Have some experience on an OS/390 platform</Prereq>
    </Prereq_list>
</Course>
```

XML Extender: Introducing XML

Structure of an XML document is defined: Schema / DTD

Document Type Definition (DTD)

```
<!ELEMENT Course (Title,Description,Prereq_list)>
<!ATTLIST Course id CDATA #REQUIRED>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Prereq_list (Prereq+)>
<!ELEMENT Prereq (#PCDATA)>
```

```
<Course id=”1”>
    <Title>DB2 UDB for OS/390<\Title>
    <Description>This 4 day course will introduce you in a new IT domain ... </Description>
    <Prereq_list>
        <Prereq>Know the concepts of a relational DataBase</Prereq>
        <Prereq>Know some basic SQL</Prereq>
        <Prereq>Have some experience on an OS/390 platform</Prereq>
    </Prereq_list>
</Course>
```

XML Extender: Features

A table: db2xml.DTD_REF is supplied:

- To store your DTD documents
- To VALIDATE inserted documents

XML Extender: Features

Store an entire XML document:

- XMLVarchar
- XMLCLOB
- XMLFILE (external file)

and extract some data for rapid search. (XColumn)

Decompose an XML-document (XCollection)

- Extract data from XML-document
- Distribute the data in existing DB2 tables

Compose an XML-document from DB2 data (XCollection)

XML Extender: Features

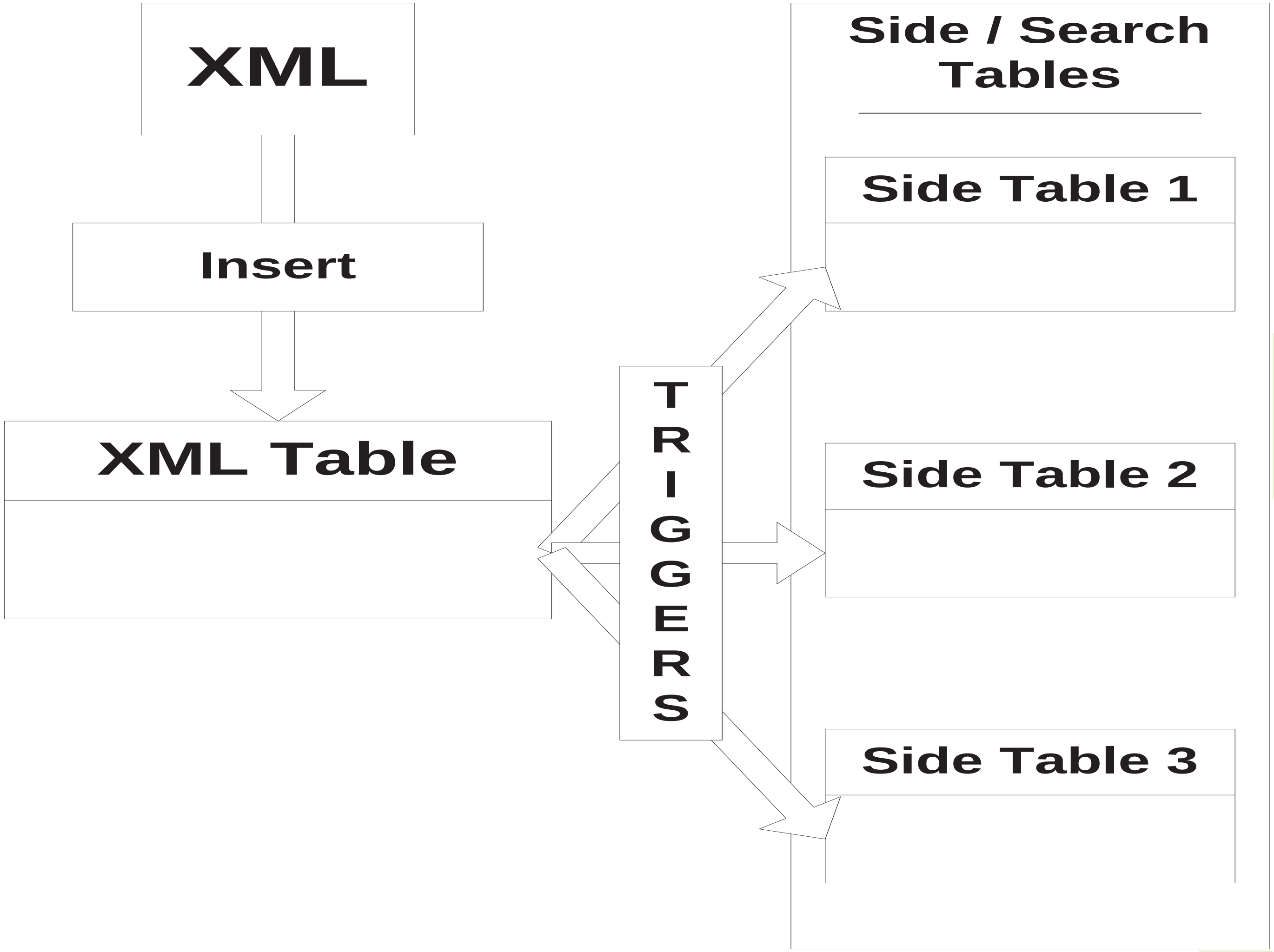
XColumn

- **Store an entire XML document**
- **Validate the XML document before insert**
- **Copy elements and attributes to side tables**
  - **For fast searching**
  - **Can be indexed**

When to use XColumn?

- (almost) read only XML documents
- Archiving or Auditing purposes
- The XML document has to stay intact!

XML Extender: Features



XML Extender: Features

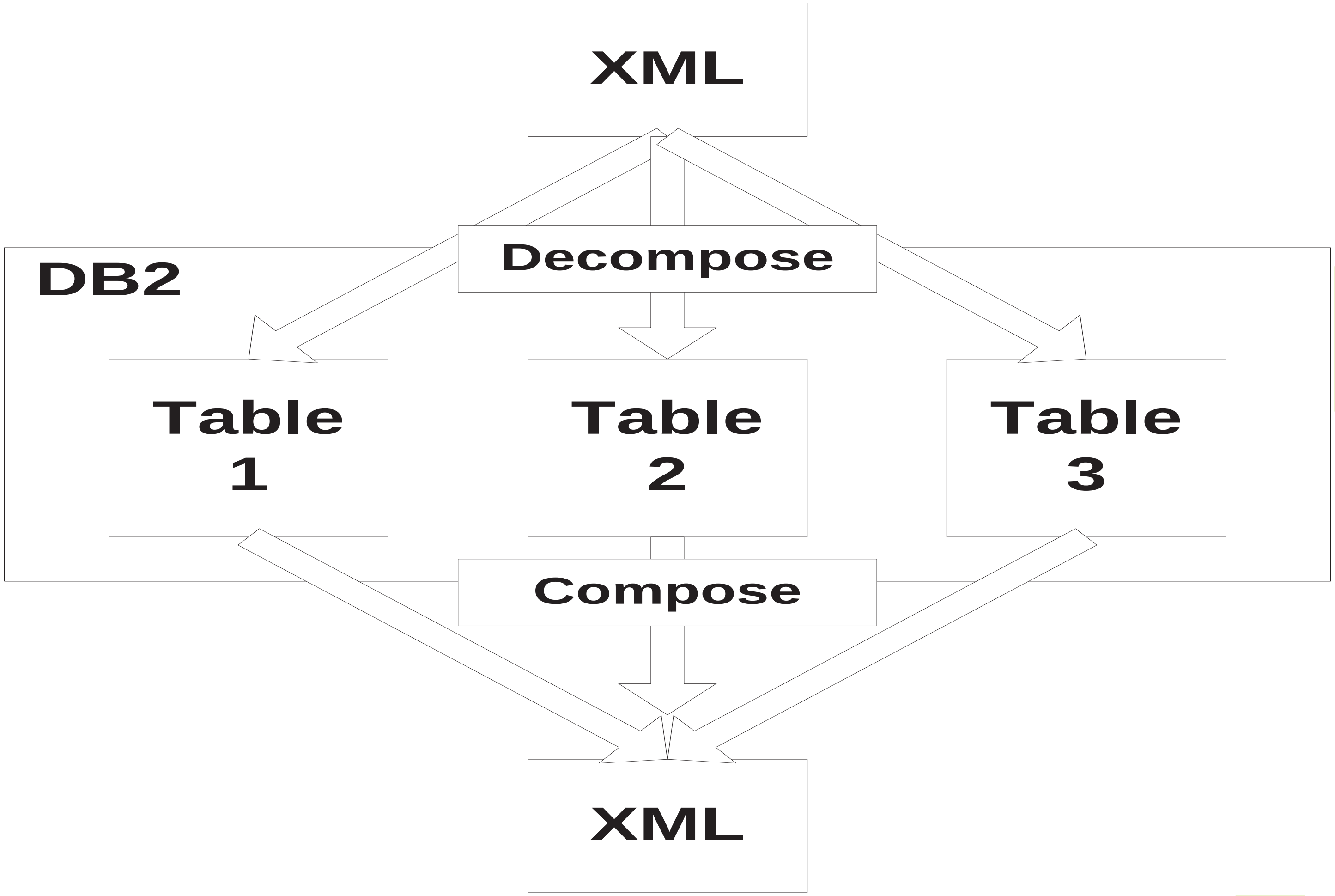
XCollection

- Validate XML document before decomposing
- Decompose XML documents and distribute XML data to existing DB2 tables (= NO XML IN TABLES)
- Compose XML documents using DB2 data and SQL

When to use XCollection?

- Your XML documents are used to transfer data, you don’t want to modify database definitions.
- Your XML data is updated frequently.
- You want to use SQL to compose your XML documents.

XML Extender: Features



XML Extender: DB2 UDB Implementation

Document Access Definition (DAD)

- XML document.
- Indicates if DB2 needs to validate.
- Indicates the method (XColumn / XCollection).
- Maps XML to Relational data.

```
<DAD>
<dtdid>g:\DDUG\DDUGXColumn\XMLdocs\CourseDesc.dtd</dtdid>
<validation>YES</validation>
<XColumn/>
</DAD>
```

XML Extender: DB2 UDB Implementation

XColumn: A Column Contains an entire XML document

- **Needs an XML UDT**
  - XMLvarchar
  - XMLCLOB
  - XMLFILE
- **Needs a DAD with:**
  - Validation: NO / YES (dtdid)
  - Side table definitions
  - Mapping between side table columns and XML data (Location Path)

After enabling XColumn, DB2 will:

- **Create all Side Tables + Default view**
- **Create triggers**
- **Add referential columns**

XML Extender: DB2 UDB Implementation

DB2 UDB Extenders: XML & Text Extender

Example: XColumn DAD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DAD SYSTEM "f:\dxx\dtd\dad.dtd">
<DAD>
    <dtdid>g:\DDUG\DDUGXColumn\XMLdocs\CourseDesc.dtd</dtdid>
    <validation>YES</validation>
    <Xcolumn>
        <table name="Side_Course_table">
            <column name="course_id" type="integer" path="/Course/@id"
multi_occurrence="NO"/>
            <column name="course_title" type="character(20)" path="/Course/Title"
multi_occurrence="NO"/>
        </table>
        <table name="Side_Prereq_table">
            <column name="course_prereq" type="varchar(60)"
path="/Course/Prereq_list/Prereq" multi_occurrence="YES"/>
        </table>
    </Xcolumn>
</DAD>
```

XML Extender: DB2 UDB Implementation

XCollection: No XML in your Database, only XML data inserted

- **Enable Collection with a DAD (optional)**
- **No preparation needed at Table / Column level**
- **A DAD / Collection has to be mentioned at each**
  - **Insert operation: dxxShredXML() / dxxInsertXML()**
    - Validation: NO / YES (dtdid)
    - Mapping between XML and RDB (RDB.DAD)
  - **Retrieve operation: dxxGenXML() / dxxRetrieveXML()**
    - Mapping between RDB and XML (RDB or SQL mapping)

XML Extender: DB2 UDB Implementation

Example: XCollection RDB_node mapping DAD

RDB_node Condition used for FK/PK relationships

```
<element_node name="Seminar">
        <RDB_node>
            <table name="seminars"/>
            <table name="speakers"/>
            <condition>speakers.seminar_id=seminars.id</condition>
        </RDB_node>
```

XML Extender: DB2 UDB Implementation

Element and Attribute Values Inserted in Columns

```
<attribute_node name="id">
    <RDB_node>
        <table name="seminars"/>
        <column name="id" type="integer"/>
    </RDB_node>
</attribute_node>


    <element_node name="Person" multi_occurrence="YES">
        <element_node name="LastName">
            <text_node>
                <RDB_node>
                    <table name="persons"/>
                    <column name="lastname" type="character(20)"/>
                </RDB_node>
            </text_node>
        </element_node>
```

XML Extender: DB2 UDB Implementation

XCollection: SQL mapping

```
<Xcollection>
    <SQL_stmt>select seminars.id as se_id, name, city, speakers.id as sp_id, firstname, lastname
                    from speakers, seminars
                    where seminars.id = seminar_id
                        and company_name = 'ABIS'
                    order by se_id, sp_id
    </SQL_stmt>
```

XML Extender: DB2 UDB Implementation

Contents of Elements & Attributes is derived from Selected Columns

```
<element_node name="Seminar">
    <element_node name="Name">
        <text_node>
            <column name="name" type="character(20)"/>
        </text_node>
    </element_node>



    <element_node name="Person" multi_occurrence="YES">
        <element_node name="FirstName">
            <text_node>
                <column name="firstname" type="character(20)"/>
            </text_node>
        </element_node>
```

DB2 UDB Extenders: XML & Text Extender

XML Extender: Enabling the XML Extender / Demo

Using the XML Extender Administration Interface,

> dxxadm ...

Prepare (enable) your database for XML Extender use.

> dxxadm enable_db “db-name”

z/OS OS/390 >> dxxadm enable_server -a “ssid” ...

Insert a DTD in the DTD_REF table for validation.

XML Extender: Enabling the XML Extender / Demo

XColumn

Enable an XML Column with a DAD

> dxxadm enable_column “database-name” “table-name” “column-name” “dad-location”

Use regular SQL to insert XML documents

XML Extender: Enabling the XML Extender / Demo

XCollection

Use the stored procedures and a DAD to decompose / compose your XML

OR

Use the sample Programs:

- **Decompose**

```
> shred “database-name” “dad-location” “xml-location”
```

- **Compose**

```
> tests2x “database-name” “dad-location” “xml-result-table”
```

Text Extender For XML documents: Enabling and Demo

Enable the Text Extender

Modify the desmodels.ini (Use the XML Location Path)

```
; list of document models
[MODELS]
modelname = Course

[Course]
Course = /Course
Content = /Course/Content
HLContent = /Course/Content/HLContent
```

Enable Column “Sections-enabled” and mention the model

Use the UDF’s to Query:

```
> db2 select db2tx.no_of_matches(hfile, 'model Course sections (HLContent) \"storage\"")
  from xmlfile
```

More Information about DB2 Extenders?

Pieter Bedert, ABIS Training and Consulting

e-mail: pbedert@abis.be

ABIS Nieuwsbrief: Exploring DB2

http://www.abis.be/nexploredb2/nexploredb2.html

ABIS UP-TO-DATE seminar

Integrating IBM technologies: Web Services (Woerden 12-13/12/2003)

http://www.abis.be/ncourses/N_UTDTOC.html