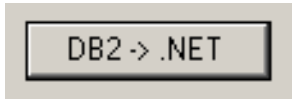
.NET and DB2 united with IBM DB2 .NET Data Provider

Objectives :

- **.NET**
- **ADO.NET**
- **DB2 and ADO.NET**
- **DB2 - ADO.NET applications**

DEMO

Win Forms client application queries DB2



according to a parameterized query

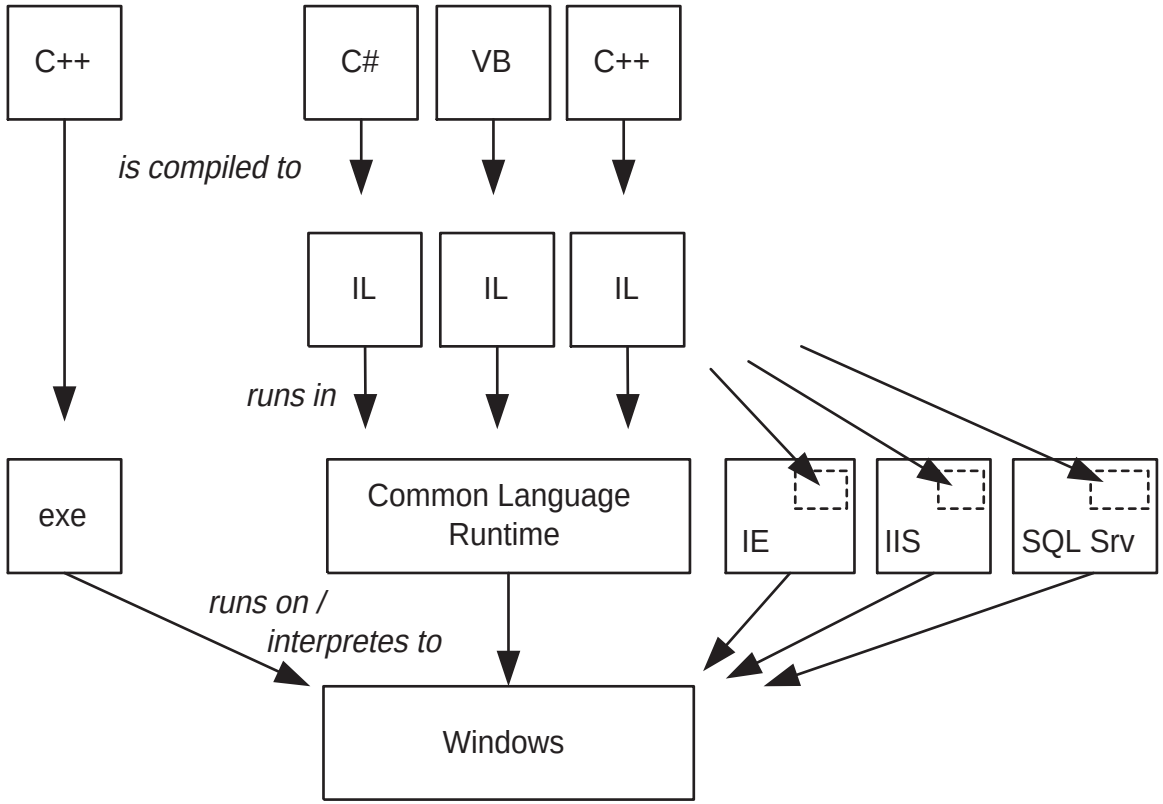
Result is displayed and changed by user

Changes are executed in DB2 database

abis
TRAINING & CONSULTING

Common Language Runtime (CLR) + Class Library

Common Language Runtime 1.1



The .NET Framework: CLR

1. Compilation and Execution Model

- source code: VB, C#, C++, COBOL, ...
- result of compilation:

  binary files containing an intermediate representation of the source constructs

  combination of metadata and Microsoft Intermediate Language (MSIL)

  In order for these binaries to be executed, the CLR must be present on the target machine
- execution:

  when these binaries are executed they cause the CLR to load

  the CLR then takes over and manages the execution

  JIT compilation (converting the MSIL as needed into the correct stream of instructions for the underlying processor)

The .NET Framework: CLR

2. The Common Type System (CTS)

- **The CTS defines the rules by which all types are declared, defined and managed, regardless of source language.**
- **The CTS is designed to be rich and flexible enough to support a wide variety of source languages**

3. The Common Language Specification (CLS)

- **interacting with code that is written in a different programming language**

The .NET Framework: Class Library

The Framework Class Library

1.2

Any programming language (conform to CLS) can use classes

Class Library includes more than 3500 classes with support for:

- **interaction with databases**

  manipulating tabular and tree-structured data

  consuming and producing XML
- **core functionality, such as interacting with base data types; console, network and file I/O, exceptions**
- **web-based (thin client) applications**
- **desktop-based (thick client) applications with broad support for the Windows GUI.**
- **SOAP-based XML web services.**

The .NET Framework: Class Library

1. Namespace: a logical design-time naming convenience

```
...
using System.Windows.Forms;
namespace WindowsApplication1
{
        public class Form1 : System.Windows.Forms.Form
    {
        private TextBox textBox1;
        private System.Windows.Forms.Button button1;
        ...
```

- class library is comprised of namespaces
- each namespace contains types that you can use in your program
- 'using' keyword (C#): to avoid fully qualified names (*name-space.typename*)
- Namespaces provide scope: Two classes with the same name can be used in your program as long as they are in different namespaces and as long as you qualify the names to the namespaces

The .NET Framework: Class Library

2. Managed DLLs (=Assemblies)

- the class library is contained in multiple DLLs or assemblies
- types in namespaces reside in a managed DLL
- namespaces organize the objects defined in an assembly
- every type is loaded in the context of an assembly
- if a certain type is needed ---> add a reference to the DLL that contains that type

Data Providers 2.1

ADO.NET includes classes to access and work with data

- **classes can be data source independent**

  ex. DataSet

  - is part of **System.Data namespace**
  - is found in **System.Data.dll**

- **classes can be data source dependent**

  key objects:

  - Connection
  - Command
  - DataReader
  - DataAdapter

  ---> provided by a .NET Data Provider

  ---> part of another namespace, other dll needed

ADO.NET: Data Providers

ADO.NET ships with a

- **OLE DB .NET Data Provider**

  = provider for OLE DB data sources

  calls through the OLE DB layer
- **SQL Server .NET Data Provider**

  communicates with native data transfer protocol of SQL Server

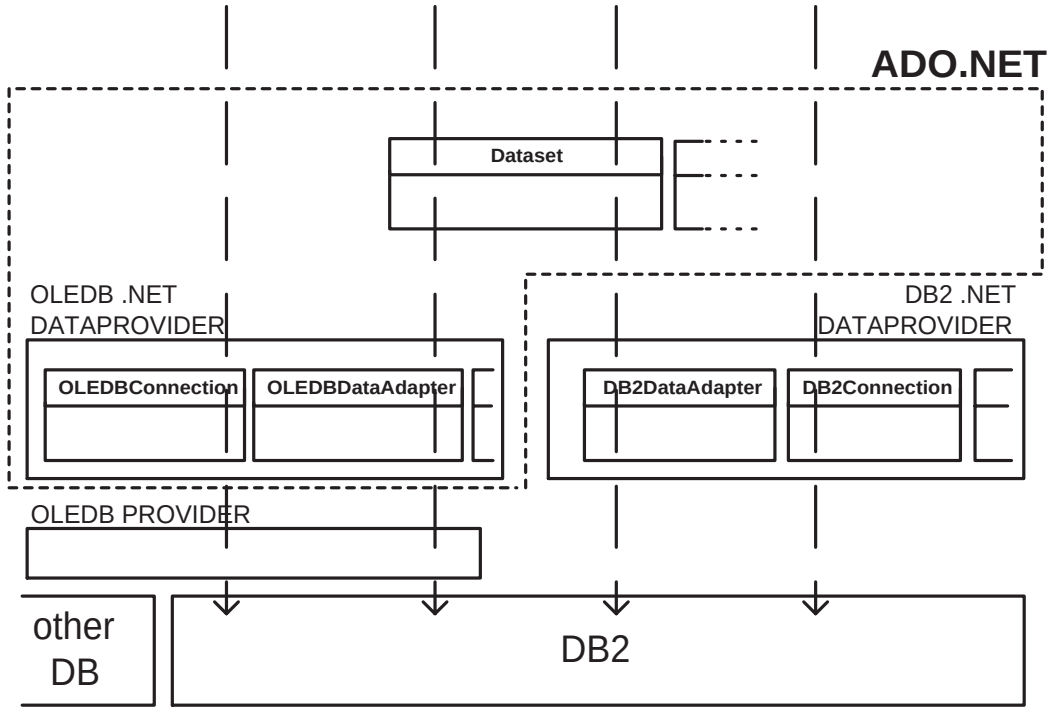
Downloadable from Microsoft: ODBC.NET Data Provider,...

You can write your own Data Provider for any source ...

native data providers replace the need of a OLE DB driver

ADO.NET: Data Providers

ADO.NET and the DB2 .NET Data Provider



ADO.NET: Architecture

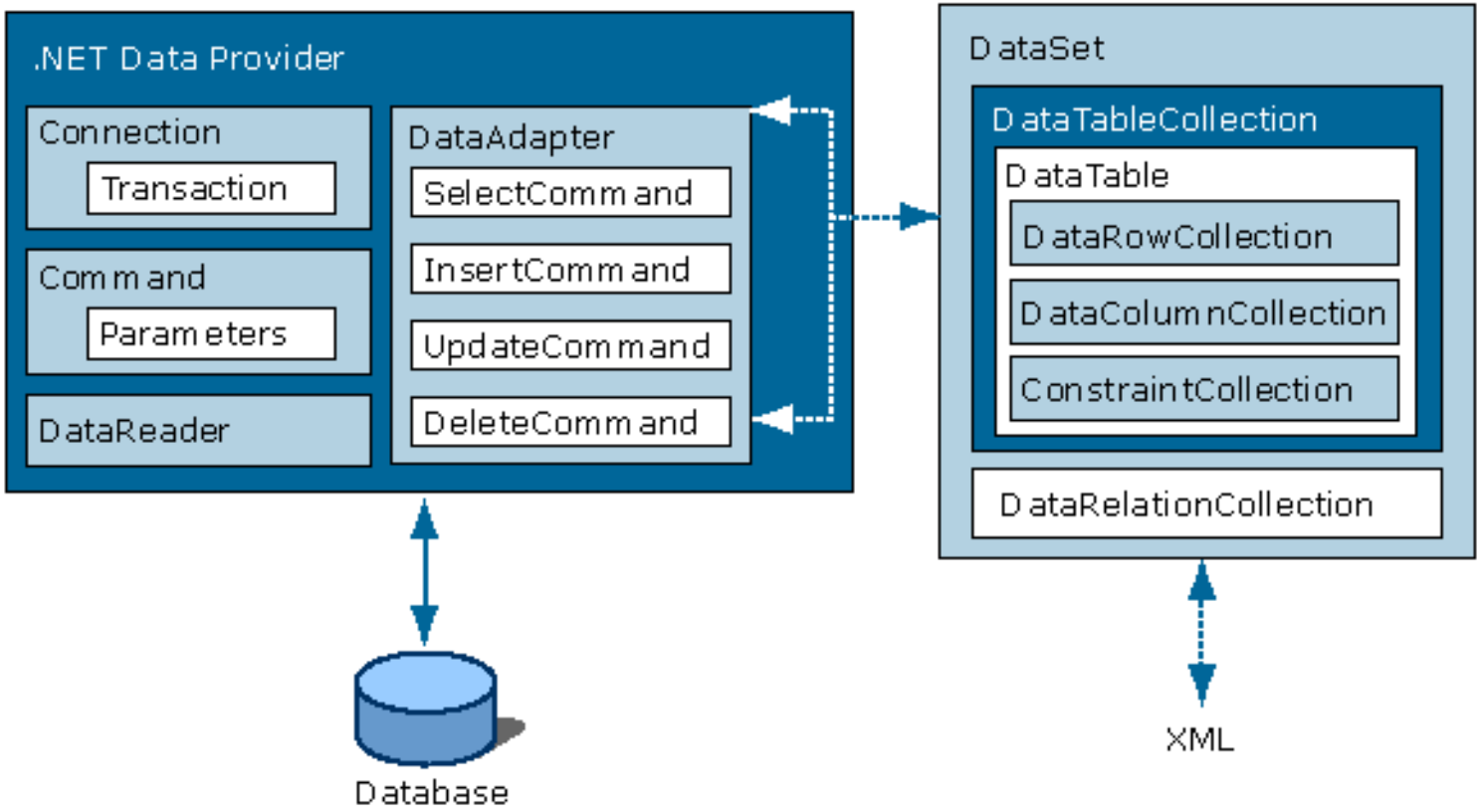
Architecture

2.2

- no continuously live connections
- tight integration with XML
- DataSet: in-memory disconnected cache of data retrieved from a da-tasource

  ==> new application architecture: loosely coupled applications
- <---> data commands for direct database access
- transactions

ADO.NET: Architecture



ADO.NET: DataSet

DataSet

2.3

- **a cache of records retrieved from a datasource**
- **independent of datasource**
- **a relational view of data that can be represented in XML**

  access and manipulate the data two ways:

  - **as tables in a relational database**
    - one table or more tables, inclusive relationships between the tables
    - constraints (unique constraints, FK constraints)
  - **as XML**
    - data is transferred and persisted as XML
- **DataSet is defined with XML schema (typed DataSet)**
- **no current record, all records are available**
- **changes to DataSet can be written back to data source**
- **diffgram**

IBM DB2.NET enablement 3.1

- DB2 development add-ins for Visual studio
- data provider

DB2 & ADO.NET: IBM data provider

DB2 .NET data provider

3.2

.NET applications can access

- DB2 Universal Database for LUW Version 8
- DB2 Universal Database for OS/390 and z/OS Version 6 and Version 7 via DB2 Connect

reference to assembly: IBM.Data.DB2.dll

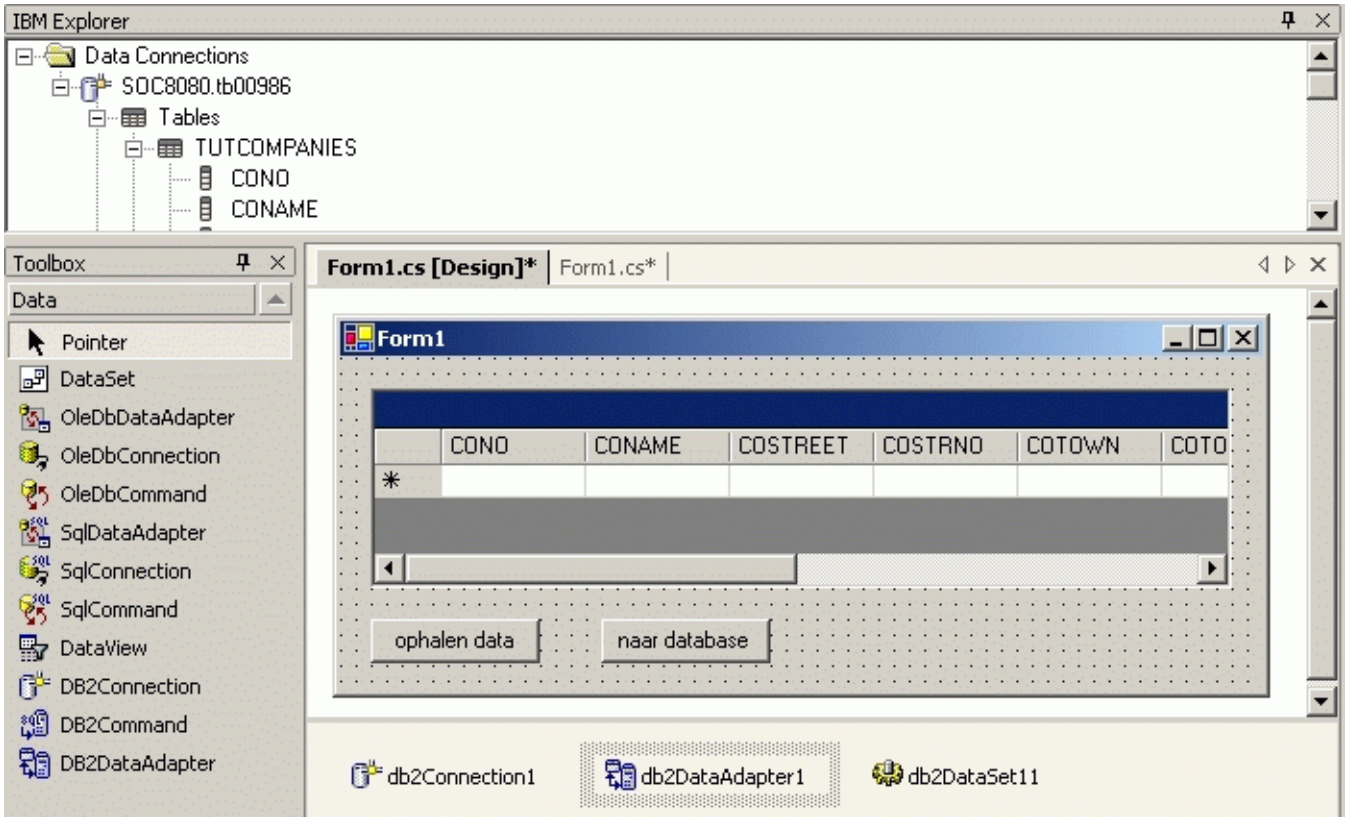
namespace: IBM.Data.DB2

Some important classes:

DB2DataAdapter
DB2Command
DB2Connection
DB2DataReader
...

Visual studio add-ins

3.3



DB2 & ADO.NET: Visual studio add-ins

- IBM Explorer
- DB2 Tools toolbar
- DB2 Database Project
- DB2 SQL Editor
- Integrated help
- Adapted Toolbox
- IBM DB2 output message pane
- DB2 Data Adapter Configuration Wizard
- generate DB2 ADO.NET data components by drag and drop
- browse metadata of DB2 objects in VS properties browser

requirements:

Visual Studio .NET 2002 + DB2 Application Development client V8.1.2

.NET 2003 ----> DB2: service pack 3 or higher

abis TRAINING & CONSULTING

Where is my cursor? 4.1

there are no updateable server-side cursors

- **DB2DataReader: forward-only, read-only, server side cursor**
- **DataSet: disconnected client cursor + DB2DataAdapter**

DB2 ADO.NET Applications: DataSet and DB2DataAdapter

Working with a DataSet and DB2DataAdapter

4.2

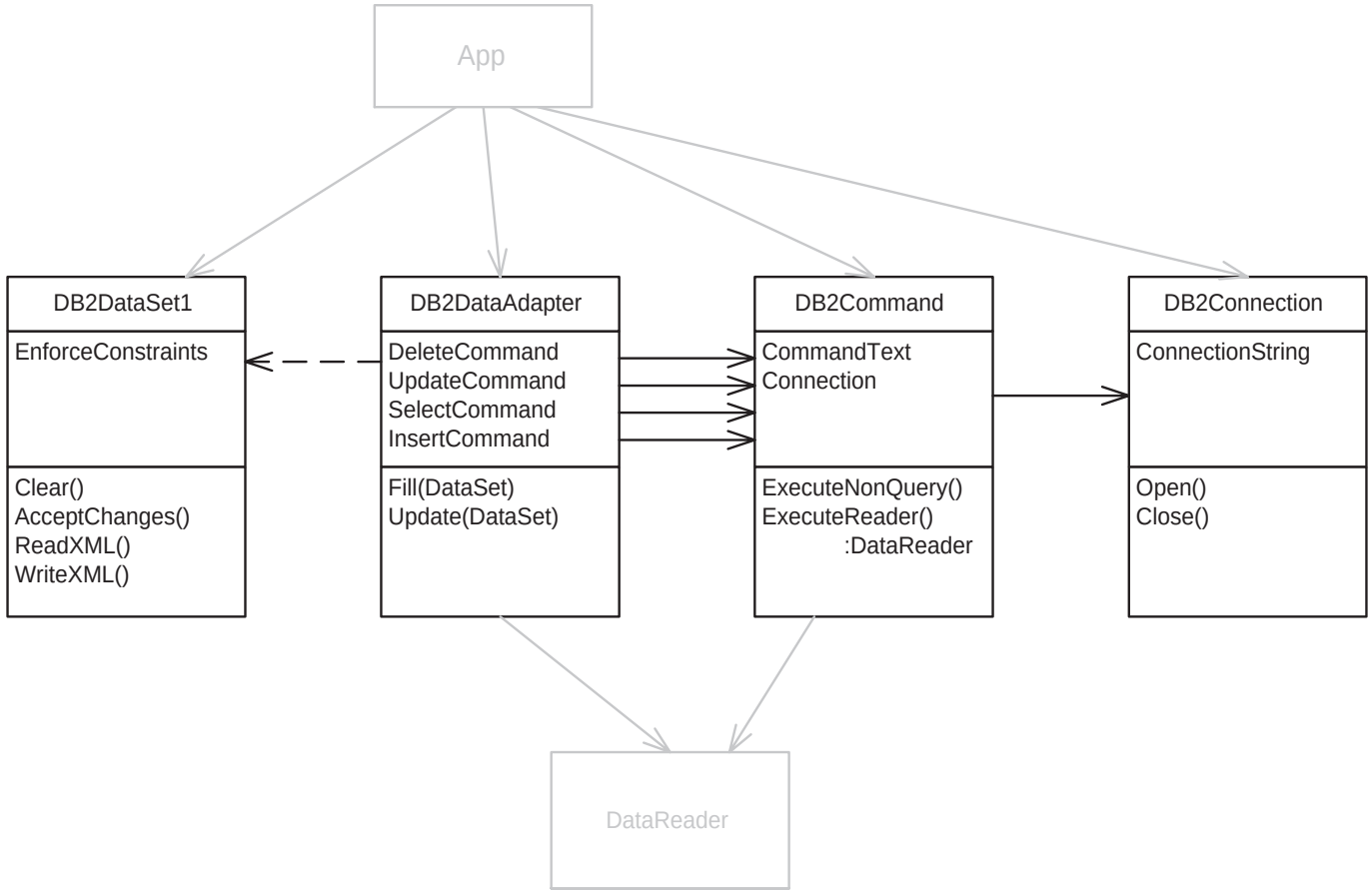
DB2DataAdapter:

- is used to exchange data between a DataSet and a DB2 table.
- represents a set of data commands and a connection to a data-base
- DataSet / DB2DataAdapters: one to many.
  Each DataAdapter can be used for one or more DataTable objects

DB2 ADO.NET Applications: DataSet and DB2DataAdapter

Fill method:

- loads data into the DataSet by executing a DB2SelectCommand



DB2 ADO.NET Applications: DataSet and DB2DataAdapter

```
private IBM.Data.DB2.DB2Command db2SelectCommand1;
private IBM.Data.DB2.DB2DataAdapter db2DataAdapter1;
private WindowsApplication2.DB2DataSet1 db2DataSet11;
private IBM.Data.DB2.DB2Connection db2Connection1;
...
this.db2SelectCommand1 = new IBM.Data.DB2.DB2Command();
this.db2Connection1 = new IBM.Data.DB2.DB2Connection();
this.db2DataAdapter1 = new IBM.Data.DB2.DB2DataAdapter();
this.db2DataSet11 = new WindowsApplication2.DB2DataSet1();
this.button1.Click += new System.EventHandler(this.button1_Click);

this.db2DataAdapter1.SelectCommand = this.db2SelectCommand1;
this.db2SelectCommand1.CommandText = "SELECT \"CONO\", \"CONAME\", ...
FROM \"DB2ADMIN\".\"TUTCOMPANIES\"
this.db2SelectCommand1.Connection = this.db2Connection1;

private void button1_Click(object sender, System.EventArgs e)
        {
            db2DataAdapter1.Fill(db2DataSet11);
        }
```

DB2 ADO.NET Applications: DataSet and DB2DataAdapter

Update method:

- **sends changes back to the database by executing DB2InsertCommand, DB2UpdateCommand and DB2deleteCommand**

```
...
this.db2DataAdapter1.UpdateCommand = this.db2UpdateCommand1;
this.db2DataAdapter1.InsertCommand = this.db2InsertCommand1;
this.db2DataAdapter1.DeleteCommand = this.db2DeleteCommand1;

this.db2UpdateCommand1.CommandText = ...
this.db2InsertCommand1.CommandText = ...
this.db2DeleteCommand1.CommandText = ...

this.db2UpdateCommand1.Connection = this.db2Connection1;
this.db2InsertCommand1.Connection = this.db2Connection1;
this.db2DeleteCommand1.Connection = this.db2Connection1;

private void button1_Click(object sender, System.EventArgs e)
        {
            db2DataAdapter1.Update(db2DataSet11);
        }
```

DB2 ADO.NET Applications: DataSet and DB2DataAdapter

Remarks:

- **updates to the datasource are performed in a 'certain order'**
  - updates and inserts of PK
  - updating child - parent tables
  - autoincrement columns
  - exceptions
- **1 sql-statement = 1 transaction**
  - stops executing when error is encountered
  - constraint, null, datatype exceptions solved in DataSet
    but DataSet contains only part of table
- **meanwhile an update can be performed at the datasource**
  - optimistic concurrency

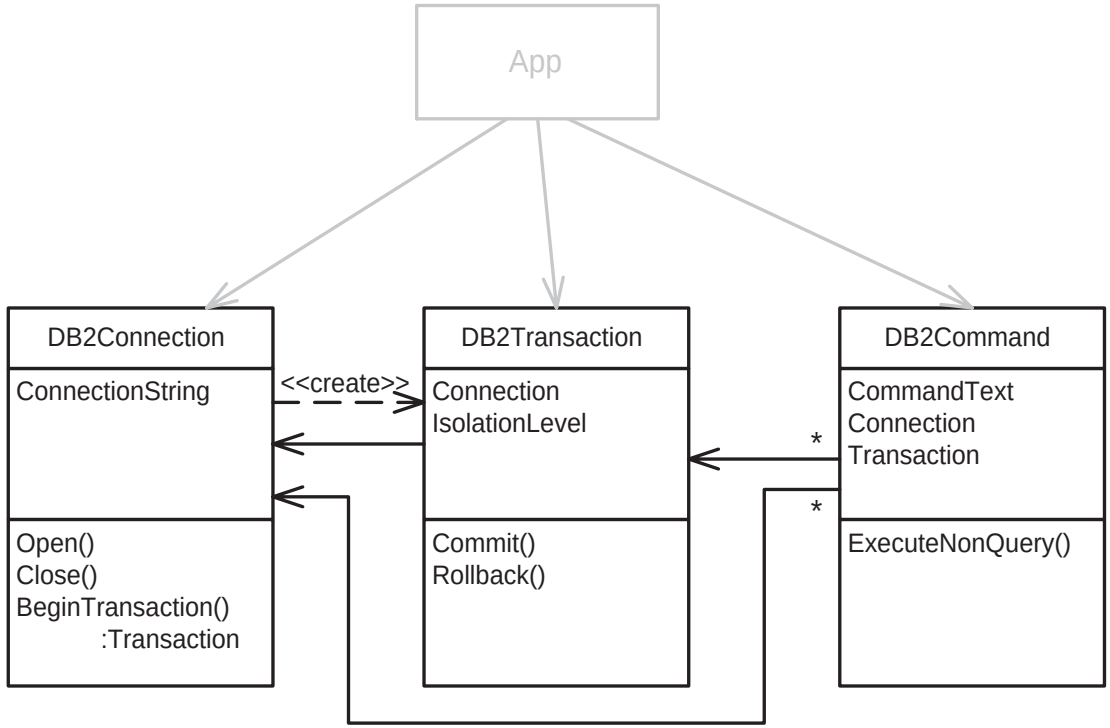
DB2 ADO.NET Applications: DataSet and transactions

DataSets and transactions

4.3

1. Working with transactions

- You must explicitly open en close the connection



DB2 ADO.NET Applications: DataSet and transactions

```
private IBM.Data.DB2.DB2Transaction db2Transaction1;
...
private void button2_Click(object sender, System.EventArgs e)
        {   try
            {   db2Connection1.Open();
                db2Transaction1 = db2Connection1.BeginTransaction();
                db2UpdateCommand1.Transaction = db2Transaction1;
                db2InsertCommand1.Transaction = db2Transaction1;
                db2DeleteCommand1.Transaction = db2Transaction1;
                db2DataAdapter1.Update(db2DataSet11);
                db2Transaction1.Commit();
                db2Connection1.Close();
            }
            catch (IBM.Data.DB2.DB2Exception exep)
            {   MessageBox.Show(exep.Message, exep.GetType().ToString());
                db2Transaction1.Rollback();
                db2DataAdapter1.Fill(db2DataSet11);
                db2Connection1.Close();
            }
        }
```

DB2 ADO.NET Applications: DataSet and transactions

2. Working without transactions

idea:

- skip update that generates the error
- notify user
- continue with next rows

```
this.db2DataAdapter1.ContinueUpdateOnError = true;
this.db2DataAdapter1.RowUpdated += new
IBM.Data.DB2.DB2RowUpdatedEventHandler(this.db2DataAdapter1_RowUpdated);
private void db2DataAdapter1_RowUpdated(object sender,
IBM.Data.DB2.DB2RowUpdatedEventArgs e)
    {
        //if (e.Status == UpdateStatus.ErrorsOccurred)
        if (e.RecordsAffected == 0)
        {
            e.Row.RowError = "Fout";
            MessageBox.Show("in RowUpdatedevent      : " + e.Errors.ToString());
        }
    }
```

DB2 ADO.NET Applications: DataSet and transactions

```
this.db2DataAdapter1.RowUpdated += new
IBM.Data.DB2.DB2RowUpdatedEventHandler(this.db2DataAdapter1_RowUpdated);

private void db2DataAdapter1_RowUpdated(object sender,
IBM.Data.DB2.DB2RowUpdatedEventArgs e)
    {
        if (e.RecordsAffected == 0)
        {
            e.Row.RowError = "Fout";
            e.Status = UpdateStatus.SkipCurrentRow
            MessageBox.Show("in RowUpdatedevent      : " + e.Errors.ToString());
        }
    }
```

DB2 ADO.NET Applications: DataSet and transactions

3. Isolation levels:

.NET	DB2
ReadCommitted	Cursor Stability (default)
ReadUncommitted	Uncommitted Read
RepeatableRead	Read Stability
Serializable	Repeatable Read

```
db2Transaction1 = db2Connection1.BeginTransaction(IsolationIevel.ReadCommitted);
```

Remark:

The DataSet has its own API, for example:

it's possible to execute updates,...,'commit' changes on the DataSet it-self

DB2 ADO.NET Applications: Optimistic Concurrency

Optimistic Concurrency

4.4

Generated update statement:

```
UPDATE "DB2ADMIN"."TUTCOMPANIES"
SET "CONO" = ?, "CONAME" = ?, "COSTREET" = ?, "COSTRNO" = ?, ...
WHERE ("CONO" =  ?) AND
("CONAME" =  ? OR CAST(? as CHARACTER(45)) IS NULL AND "CONAME" IS NULL)AND
("COSTREET" =  ?) AND
("COSTRNO" =  ? OR CAST(? as VARCHAR(10)) IS NULL AND "COSTRNO" IS NULL) AND
("COTOWN" =  ? OR CAST(? as CHARACTER(45)) IS NULL AND "COTOWN" IS NULL) ...
```

two techniques:

DBConcurrencyException:

The exception that is thrown by the DataAdapter during the update operation if the number of rows affected equals zero.

or Rowupdated event

DB2 ADO.NET Applications: Optimistic Concurrency

```
private void button2_Click(object sender, System.EventArgs e)
        {
            try
            {
                db2DataAdapter1.Update(db2DataSet11);
            }
            catch (DBConcurrencyException exep)
            {
                MessageBox.Show("foutDBconcurrencyException");
            }
        }
```

DB2 ADO.NET Applications: DB2CommandBuilder

DB2CommandBuilder

4.5

Automatically generates single-table commands

uses the SelectCommand property to retrieve a required set of meta-data

The DB2CommandBuilder also uses the Connection, CommandTime-out, and Transaction properties referenced by the SelectCommand

```
private IBM.Data.DB2.DB2CommandBuilder db2CommandBuilder1;
...
this.db2DataAdapter1.SelectCommand = this.db2SelectCommand1;
this.db2CommandBuilder1 = new IBM.Data.DB2.DB2CommandBuilder();
this.db2CommandBuilder1.DataAdapter = this.db2DataAdapter1;
```

DB2 ADO.NET Applications: DB2CommandBuilder

Trace output:

- dynamic prepare select command
- call SYSIBM.SQLPRIMARYKEYS
- generated update statement

```
UPDATE "DB2ADMIN"."TUTCOMPANIES"
SET  "CONAME" = ?
WHERE ("CONO" =  ?) AND
("CONAME" =  ? OR CAST(? as CHARACTER(45)) IS NULL AND "CONAME" IS NULL)AND
("COSTREET" =  ?) AND
("COSTRNO" =  ? OR CAST(? as VARCHAR(10)) IS NULL AND "COSTRNO" IS NULL) AND
("COTOWN" =  ? OR CAST(? as CHARACTER(45)) IS NULL AND "COTOWN" IS NULL) ...
```

abis TRAINING & CONSULTING

Questions?

kplatteborze@abis.be