Module 17

"Accessing Data"

TEKNOLOGISK
INSTITUT

TEKNOLOGISK
INSTITUT

Agenda

- Connected Access
- Disconnected Access
- LINQ to XML

TEKNOLOGISK
INSTITUT

Using ADO.NET Providers

- .NET 4.5 contains
  - OleDb
    - Access
    - SQL Server 6.5 and earlier
    - …
  - Odbc
    - MySql
    - …
  - SQL Server
    - SQL Server 7.0 and later
- Oracle now supplies its own .NET provider
  - Download from Oracle's web site

TEKNOLOGISK INSTITUT

ADO.NET Provider Classes

- `IDbConnection` — Generic interface
  - `DbConnection` — Base class
    - `SqlConnection` — SqlClient
- `IDbCommand` — Generic interface
  - `DbCommand` — Base class
    - `SqlCommand` — SqlClient
- `IDataReader` — Generic interface
  - `DbDataReader` — Base class
    - `SqlDataReader` — SqlClient

SqlConnection

TEKNOLOGISK INSTITUT

- SqlConnection
  - Open()
  - Close()
  - Dispose()

```
string cs = @"data source=localhost; Integrated Security=SSPI;
   Database=...;";
using( SqlConnection connection = new SqlConnection( cs ) )
{
   connection.Open();

   // ...
}
```

- Connection string can be read from configuration file

SqlCommand

TEKNOLOGISK INSTITUT

- SqlCommand
  - CommandText
  - CommandType
  - **ExecuteScalar()**
  - **ExecuteNonQuery()**
  - **ExecuteReader()**

```
SqlCommand command = connection.CreateCommand();
command.CommandType = CommandType.Text;
command.CommandText =
   "UPDATE People SET FirstName = 'Hidi' WHERE Id = 7";
connection.Open();

int count = command.ExecuteNonQuery();
```

- Exam tip: Remember to open connection first!

SqlDataReader

TEKNOLOGISK INSTITUT

- Provides forward-only, read-only server side cursor
- SqlDataReader
  - Read()
  - NextResult()

```
using ( SqlDataReader reader = command.ExecuteReader() )
{
   while( reader.Read() )
   {
      Console.WriteLine(
         string.Format( "{0}: {1} {2} ",
           reader[ "id" ],
           reader[ "FirstName" ],
           reader[ "LastName" ] ) );
    }
}
```

TEKNOLOGISK
INSTITUT

Agenda

- Connected Access
- **Disconnected Access**
- LINQ to XML

TEKNOLOGISK
INSTITUT

ADO.NET Entity Framework

- The de-facto standard for disconnected data access providing
  - Entity Data Models (EDM)
  - Entity SQL
  - Object Services
- It supports
  - Writing code against a conceptual model
  - Type-safe data access
  - Robustness and indepedance across storage systems
  - Maintainability
- Tools and wizards supporting
  - Database-first design
  - Code-first design

TEKNOLOGISK INSTITUT

Querying and Updating Data

- Using LINQ to Entities to query data

```
using( ShopEntities entities = new ShopEntities() )
{
   var query = from c in entities.Customers
               where c.Orders.Count > 0
               select c;
   ...
}
```

- `DbContext`-generated class
  - keeps tracks of updates
  - saves back to database

```
ShopEntities entities = ...;
...
entities.SaveChanges();
```

Customizing Classes

TEKNOLOGISK INSTITUT

- Never modify the auto-generated classes!!
  - Instead, augment the auto-generated partial classes

```
public partial class Customer
{
    public string FullName
    {
        get
        {
            return FirstName + " " + LastName;
        }
    }
    public int Age
    {
        get { return ...; }
    }
}
```

TEKNOLOGISK
INSTITUT

Agenda

- Connected Access
- Disconnected Access
- LINQ to XML

TEKNOLOGISK
INSTITUT

Introducing LINQ to XML

- Provides querying facilities over XML documents
  - Introduces a new **`XDocument`** class set deriving from **`Xobject`**
  - In **`System.Xml.Linq`** namespace

- **`XAttribute`**
- XNode
  - *XContainer*
    - **`XDocument`**
    - **`XElement`**
  - XComment
  - XText
    - XCData
- ...

XDocument

TEKNOLOGISK INSTITUT

- Provides main access to XML document handling
- `XDocument.`
  - `Load()` static
  - `Parse()` static
  - `Save()`

```
XDocument doc = XDocument.Load( @"C:\Tmp\Movies.xml" );
```

```
XDocument doc = XDocument.Parse( "<Customers>...</Customers" );
```

```
doc.Save( @"C:\Tmp\CustomersOrders.xml" );
```

TEKNOLOGISK INSTITUT

Querying with LINQ to XML

- Use LINQ queries over the DOM provided by the Xdocument hierarchy classes

```
XDocument doc = XDocument.Load( @"C:\Tmp\CustomersOrders.xml" );
var query = from order in doc.Descendants( "Order" )
            where order.Attribute( "OrderID" ).Value == "10677"
            select new
            {
               OrderID = (int) order.Attribute( "OrderID" ),
               CustomerID =
                 (string) order.Parent.Attribute( "CustomerID" ),
               Freight = (decimal) order.Attribute( "Freight" )
            };
```

- The full power of LINQ is available, e.g. **join**, **group** etc.

Transforming XML to Objects

TEKNOLOGISK INSTITUT

- LINQ to XML is perfect for transforming XML
  - XML -> objects
  - XML -> text
  - XML -> XML

```
List<Customer> customersOrders =
   ( from c in doc.Descendants( "Customer" )
     select new Customer
     {
        Id = c.Attribute( "CustomerID" ).Value,
        Name = c.Attribute( "CompanyName" ).Value,
        Orders = ( from o in c.Elements( "Order" )
                   select new Order
                   {
                      Id = (int) o.Attribute( "OrderID" ),
                      Freight = (decimal) o.Attribute( "Freight" )
                   } ).ToList()
     } ).ToList();
```

TEKNOLOGISK
INSTITUT

Summary

- Connected Access
- Disconnected Access
- LINQ to XML

Question

TEKNOLOGISK INSTITUT

```
01 List<Person> persons = new List<Person>();
02 using ( SqlConnection connection = ... ) {
03 SqlCommand command = new SqlCommand( @"SELECT FirstName,
04                                LastName FROM People", connection );
05 using ( SqlDataReader reader = command.ExecuteReader() )
06 {
07    {
08       persons.Add( new Person{
09          FirstName = reader[ "FirstName" ].ToString(),
10          LastName = reader[ "LastName" ].ToString() } );
11    }
12 }
```

Which code segments needs to be added? (Choose two.)

a) Insert `connection.BeginTransaction();` before line 05

b) Insert `connection.Open();` before line 05

c) Insert `while( reader.NextResult() )` at line 07

d) Insert `while( reader.Read() )` at line 07

e) Insert `while( reader.GetValues() )` at line 07

f)

TEKNOLOGISK
INSTITUT