**Entity Framework: Yet Another ORM?**

**Basics:**

1. Intro

* Abstraction between conceptual model and database model
* How?
  + **Conceptual model:** defines entities and their relationships from the application point   
    of view. This model is defined using Conceptual Schema Definition Language (CSDL).
  + **Storage model:** defines entities and their relationships from the database point of view. This model is defined using Store Schema Definition Language (SSDL) .
  + **Mapping model:** defines how conceptual model translates into storage model. It is defined using Mapping Specification Language (MSL) .

.CDSL

.MSL

.SSDSL

Conseptual Model

Mapping

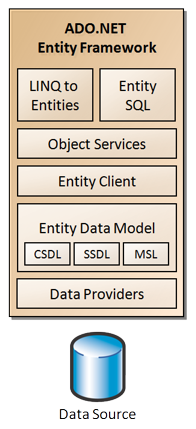
Storage Model

Application

Database

Entities

Tables



* New Features in EF 4.0
  + **Model-First Development**  
    Start from a model, and generate database based on that model
  + **Customizable Code-Generation**  
    With help of T4 code generation templating:
    - Entity Object Code Generator
    - Self Tracking Entities Code Generator
    - POCO Generator (template available online)
    - You own T4 Template
  + **Complex Types**Complex types are non-scalar properties of entity types that enable scalar properties to be organized within entities.
  + **Pluralization**  
    More meaningful names for entity types, entity sets and relationships
  + **Multi-Targeting Support**Continue building your EF 3.5 applications or move forward to EF4
  + **Persistence Ignorance & POCO**Use pure domain classes without needing to implement interfaces that deal with persistence concerns. Dynamic proxies are supported for lazy loading and efficient change tracking.
  + **Self-Tracking Entities to facilitate N-Tier applications**
  + **Application Patterns and Testability**Additional interface to help writing testable code when using the framework (IObjectSet<T>).   
    Use of patterns such as Repository and UnitOfWork are possible (MS P&P)
  + **Foreign Key Associations**FK associations allow you to include FK properties in your model and use those as a basis for relationship between entities.
  + **Lazy Loading (Enabled by default in EF 4)**
  + **More LINQ Operators in LINQ to Entities**Contains, Single, SingleOrDefault, DefaultIfEmpty, …
  + **CreateDatabase and DDL Provider Services**Allows you to do database creation based on a model.
  + **Model Defined Functions LINQ support**Define composable functions in your model using Entity SQL  
    Encapsulate some commonly used eSQL in a function.
  + **EntityDataSource support for QueryExtender, POCO and FKs**Support for ASP.NET QueryExtender and POCO entities.  
    More control over the data retrieval query of a Data Source
  + **ObjectMaterialized event**Write logic that is executed immediately after an object has been materialized
  + **Generated SQL improvements for better performance and readability**Removal of unnecessary joins, better translations of certain functions, removal of unneeded levels of nesting, and more.
  + **Ad-hoc native query support**Directly execute arbitrary data source commands.  
    The store command is executed in the context of the current transaction.
  + **…**
* What about the database?   
  … Entity Framework is database/datasource independent  
    
  2 providers ship out of the box:
  + EntityClient Provider for the Entity Framework
  + NET Framework Data Provider for SQL Server

Third party providers provide access to other databases:

* + Oracle
  + MySql
  + PostgreSQL
  + SQL Anywhere
  + DB2
  + Informix
  + … (full list on http://msdn.microsoft.com/en-us/data/dd363565.aspx)
* Querying the entity framework
  + **Entity SQL**

Is text-based, collection-oriented and late-bound query language which is influenced by Transact-SQL

* + **Linq to entities**

Is more powerful and advanced than LINQ to SQL

1. Getting started:

* Approaches
  + Database First:

Setup your database and generate the model based on the database.

* + Model First:  
    Define your model and then generate the database, mappings, and classes from the model.  
    **!!! Database will be recreated from scratch !!!** Table inheritance:
    - *Table-per-type*: Uses a separate table in storage to maintain data for each type in

the inheritance hierarchy.

* + - T*able-per-hierarchy*: Uses one table in storage to maintain data for all the types in an inheritance hierarchy.
  + Code Only (ADO.NET Entity Framework Feature CTP) (  
    Use the Entity Framework using POCO-entities and without an EDMX file.  
    Allows to write domain classes without ever looking at or touching a designer or dealing with XML.

(Create POCO, create objectcontext, setup configuration in code and you’re ready to go)

* QWars
  + - Setup
    - CRUD
    - Lazy loading / Eager fetching
    - Polymorphism
    - Performance tuning
      * Batch
      * Caching
      * Compiled queries
      * pre-generated Entity Framework view

1. Extra’s

* EdmGen
* DBGen

1. Evaluation
2. References:

* Entity Framework Design Blog: <http://blogs.msdn.com/efdesign/default.aspx>
* ADO.Net Team Blog: <http://blogs.msdn.com/adonet/default.aspx>
* Data Developer Center: <http://msdn.microsoft.com/en-us/data/default.aspx>
* Visual Studio Gallery : <http://visualstudiogallery.msdn.microsoft.com/en-us/23df0450-5677-4926-96cc-173d02752313>
* <http://thedatafarm.com/blog/>

1. Things to organize in my head … ☺

Entity Data Model:

* Designer window

Tasks:

* Create and modify entities
* Create and modify associations between entities
* View and edit mappings
* Modify inheritance relationships

Components:

* + - Designer surface: A visual surface for creating and modifying the conceptual model.
    - Mapping Details window: The location where mappings are created or modified.
    - Toolbox: Contains controls that can be used to create entities, associations, and inheritance relationships.
    - Model Browser window: Provides a view of the conceptual model and the associated storage model.

Entities / Properties / Complex types / Foreign keys & associations / Navigation properties / Mapping details

SSDL / CSDL / MSL (EntityType, Association & Mapping)

Querying (Linq to entities, method based, entity sql)

Lazy loading / eager fetching

ObjectContext / ObjectStateEntry

Tracking / Saving changes

CRUD / Stored procedures (EF Functions)

Relationships / associations / Referential constraints / Foreign keys

T4 code generation (quicky)

Updating Model

Metadata artifact processing: copy to output

Exception handling