



Manfred Steyer | CAMPUS 02 | softwarearchitekt.at

NHibernate vs. Entity Framework



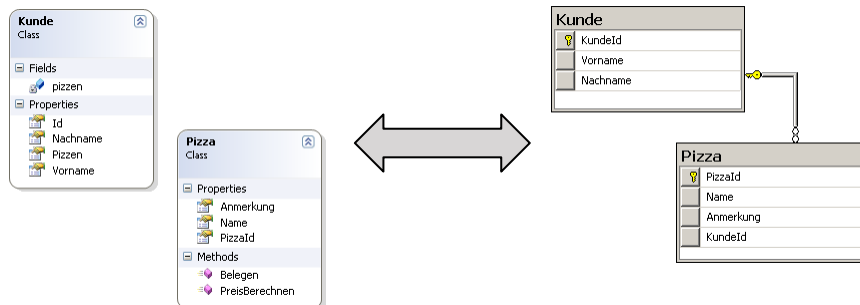
Ziele

- NHibernate und Entity Framework sowie deren Unterschiede kennen lernen

Agenda

- Kriterien
- Beispiel mit EF
- Beispiel mit NH
- Model First/ Code First
- Vererbung
- Abgehängte Objekte
- Custom SQL
- Performance

O/R-Mapping



Kriterien

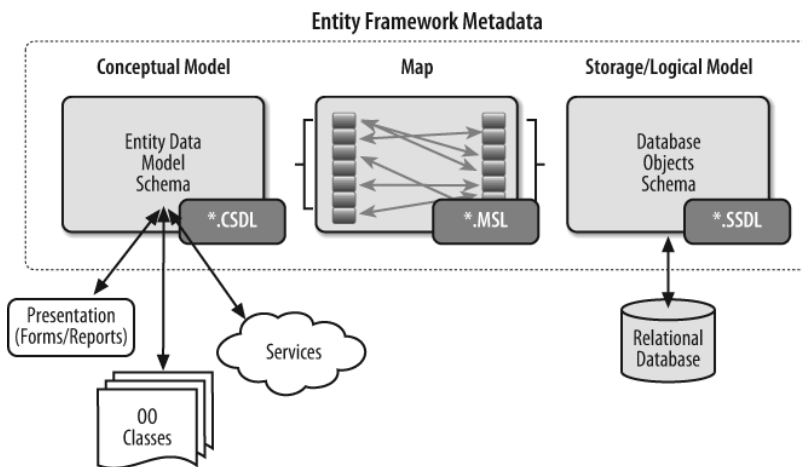
	EF	NH
Unterstützte DBMS		
Visual Studio-Integration		
Persistence Ignorance		
DB-First		
Model First/ Code First		
Abfragen		
Concurrency		
Beziehungen		
Vererbung		
Lazy-/ Eager-Loading		
Abgehängte Objekte		
Custom SQL/ SP		
Performance		

Skala

- + Ok
- ++ Perfekt
- +++ Wow! (> 100%)
- - Nicht Ok
- -- Total Schlecht
- --- Negativ Überrascht

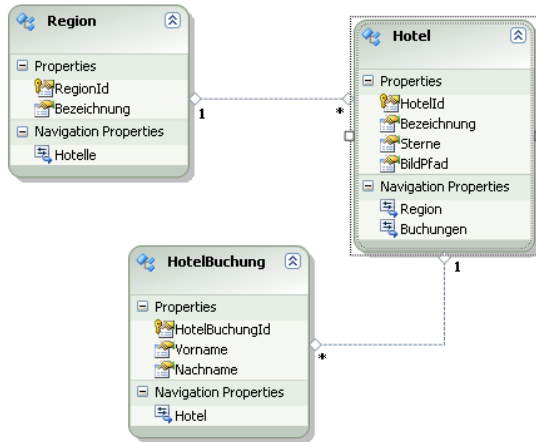
EF - ÜBERBLICK

Entity Data Model



Entnommen aus: [Lerman, Programming Entity Framework]

Entity Data Model



Entity Data Model

Mapping Details - Hotel

Column	Op...	Value / Property
Tables		
Maps to Hotel		
<Add a Condition>		
Column Mappings		
HotelId : int	↔	HotelId : Int32
Bezeichnung : varchar	↔	Bezeichnung : String
Sterne : int	↔	Sterne : Int32
RegionId : int	↔	RegionId : Int32
BildPfad : varchar	↔	BildPfad : String
<Add a Table or View>		

Generierte Artefakte

- Entitätsklassen mit Assoziationen
- ObjectContext

Unterstützte Datenbanken

- DB2
- Ingres
- PostgreSQL
- MySQL
- Oracle
- Sybase
- Oracle
- MS SQL
- Firebird
- SQL Light

DEMO

Kriterien

	EF	NH
Unterstützte DBMS		
Visual Studio-Integration		
Persistence Ignorance		
DB-First		
Model First/ Code First		
Abfragen		
Concurrency		
Beziehungen		
Vererbung		
Lazy-/ Eager-Loading		
Abgehängte Objekte		
Custom SQL/ SP		
Performance		

NH - ÜBERBLICK

XML-Mapping für Entities

```
<?xml version="1.0" encoding="utf-8" ?>
<hibernate-mapping xmlns="urn:hibernate-mapping-2.2"
  namespace="NHibernateSample.entities"
  assembly="NHibernateSample">

  <class name="Pizza">

    <id name="PizzaId">
      <column name="PizzaId" />
      <generator class="native" />
    </id>

    <property name="Bezeichnung">
      <column name="bez" />
    </property>

  </class>

</hibernate-mapping>
```


Konfiguration

```
<hibernate-configuration xmlns="urn:nhibernate-configuration-2.2">
  <session-factory>
    <property name="dialect">[...]</property>
    <property name="connection.provider">[...]</property>
    <property name="connection.connection_string">[...]</property>
    <property name="proxyfactory.factory_class">[...]</property>
    <property name="hbm2ddl.auto">create</property>
    <property name="show_sql">>false</property>

    <mapping assembly="NHibernateSample" />
  </session-factory>
</hibernate-configuration>
```

Entitäten

```
class Pizza
{
  public virtual int? PizzaId {get;set;}
  public virtual string Bezeichnung {get;set;}
  public virtual decimal Preis {get; set;}
}
```

Unterstützte DBMS

- MS SQL
- DB2
- EffiProz
- FireBird
- Informix
- MySQL
- Oracle
- PostgreSQL
- SQLite
- Sybase
- VistaDB

DEMO

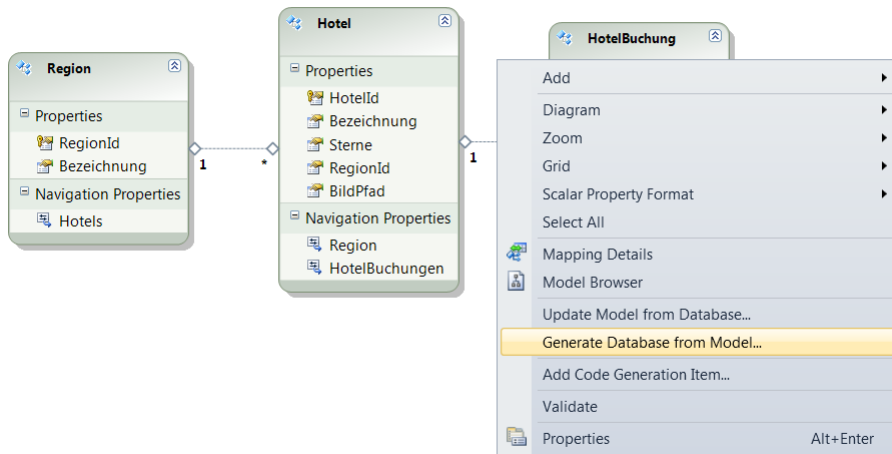
	EF	NH
Unterstützte DBMS		
Visual Studio-Integration		
Persistence Ignorance		
DB-First		
Model First/ Code First		
Abfragen		
Concurrency		
Beziehungen		
Vererbung		
Lazy-/ Eager-Loading		
Abgehängte Objekte		
Custom SQL/ SP		
Performance		

Kriterien

	EF	NH
Unterstützte DBMS	+	+++
Visual Studio-Integration	++	--
Persistence Ignorance	+	+
DB-First	++	++
Model First/ Code First		
Abfragen	++	++
Concurrency	++	++
Beziehungen	++	+++
Vererbung		
Lazy-/ Eager-Loading	++	++
Abgehängte Objekte	++	++
Custom SQL/ SP		
Performance		

EF - MODEL/ CODE FIRST

Model First



Code First (EF CTP 4 !!)

```

public class Region
{
    public int RegionId { get; set; }
    public string Bezeichnung1 { get; set; }
    public virtual ICollection<Hotel> Hotels { get; set; }
}

public class Hotel
{
    // ...
}

public class HotelContext : DbContext
{
    public HotelContext(): base("HotelDb-CodeFirst2") { }

    public DbSet<Hotel> Hotels { get; set; }
    public DbSet<Region> Regionen { get; set; }
}

```

Code First (EF CTP 4 !!)

```
public class Region
{
    [Key]
    public int RegionCode { get; set; }

    [Required]
    public string Bezeichnung { get; set; }

    [ConcurrencyCheckAttribute]
    public long Version { get; set; }
}
```

Code First (EF CTP 4 !!)

```
modelBuilder
    .Entity<Region>()
    .HasKey(r => r.RegionCode)
    .HasMany(r => r.Hotels)
    .WithRequired(h => h.Region)
    .HasConstraint( (h,r) => h.RegionId == r.RegionCode )
    .WillCascadeOnDelete();
```

NH - MODEL/ CODE FIRST

NHibernate Mapping Attributes

```
using NHibernate.Mapping.Attributes;

[Class(Name = "NHibernateSample.entities.Pizza,NHibernateSample")]
public partial class Pizza
{
    [Id(0, Name = "PizzaId")]
    [Generator(1, Class = "native")]
    public virtual int? PizzaId { get; set; }

    [Property(Name = "Bezeichnung")]
    public virtual string Bezeichnung { get; set; }

    [Property(Name = "Preis")]
    public virtual decimal Preis { get; set; }
}
```

Auto-Mapping mit Fluent NH

```
public class Pizza
{
    public int Id { get; set; }
    public int Bezeichnung { get; set; }
    public Kunde Kunde { get; set; }
}
```

Fluent NH

```
public class PizzaMap : ClassMap<Pizza>
{
    public PizzaMap()
    {
        Id(p => p.PizzaId).GeneratedBy.Native();

        Map(p => p.Bezeichnung);

        Map(p => p.Preis);

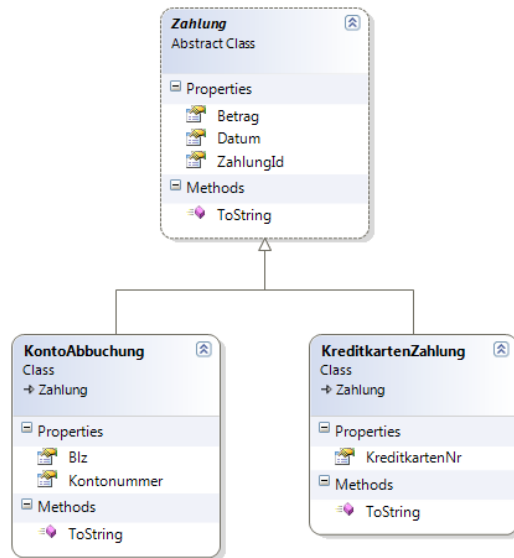
        References(p => p.Kunde).Cascade.All();

        Table("Pizzen");
    }
}
```

Kriterien

	EF	NH
Unterstützte DBMS	+	+++
Visual Studio-Integration	++	--
Persistence Ignorance	+	+
DB-First	++	++
Model First/ Code First	++	++
Abfragen	++	++
Concurrency	++	++
Beziehungen	++	+++
Vererbung		
Lazy-/ Eager-Loading	++	++
Abgehängte Objekte	++	++
Custom SQL/ SP		
Performance		

VERERBUNG



Kriterien

	EF	NH
Unterstützte DBMS	+	+++
Visual Studio-Integration	++	--
Persistence Ignorance	+	+
DB-First	++	++
Model First/ Code First	++	++
Abfragen	++	++
Concurrency	++	++
Beziehungen	++	+++
Vererbung	++	++
Lazy-/ Eager-Loading	++	++
Abgehängte Objekte	++	++
Custom SQL/ SP		
Performance		

CUSTOM SQL/ SP

EF: Native/ Store Queries

```
string sql = "select RegionID, rtrim(RegionDescription) + '!' ...";  
IEnumerable<Region> result = ctx.ExecuteStoreQuery<Region>(sql, "Region");
```

NH: SQL Queries

```
var q = session.CreateSQLQuery("select * from Pizza");
q.AddEntity(typeof(Pizza));
var pl = q.List<Pizza>();
```

Kriterien

	EF	NH
Unterstützte DBMS	+	+++
Visual Studio-Integration	++	--
Persistence Ignorance	+	+
DB-First	++	++
Model First/ Code First	++	++
Abfragen	++	++
Concurrency	++	++
Beziehungen	++	+++
Vererbung	++	++
Lazy-/ Eager-Loading	++	++
Abgehängte Objekte	++	++
Custom SQL/ SP	++	++
Performance		

PERFORMANCE

Hinterhof-Benchmark

- Performance in etwa die selbe bei kleiner Anzahl an Datensätzen
- NH ist beim Abrufen ein wenig schneller (ca. 1 ms pro Objekt)
- EF ist beim Einfügen ein wenig schneller (ca. 1 ms pro Objekt)
- Update: Ungefähr gleich (NH winzigen Vorsprung)
- Performance entspricht ~ manuellem Mapping

Zusammenfassung

	EF	NH
Unterstützte DBMS	+	+++
Visual Studio-Integration	++	--
Persistence Ignorance	+	+
DB-First	++	++
Model First/ Code First	++	++
Abfragen	++	++
Concurrency	++	++
Beziehungen	++	+++
Vererbung	++	++
Lazy-/ Eager-Loading	++	++
Abgehängte Objekte	++	++
Custom SQL/ SP	++	++
Performance	+	+

Zusammenfassung

	EF	NH
Unterstützte DBMS	+	+++
Visual Studio-Integration	++	--
Persistence Ignorance	+	+
DB-First	++	++
Model First/ Code First	++	++
Abfragen	++	++
Concurrency	++	++
Beziehungen	++	+++
Vererbung	++	++
Lazy-/ Eager-Loading	++	++
Abgehängte Objekte	++	++
Custom SQL/ SP	++	++
Performance	+	+

Literatur

- NHibernate Reference Documentation
<http://nhforge.org/doc/nh/en/index.html>
- Absolute Beginners Guide to Entity Framework
<http://blogs.msdn.com/b/adonet/archive/2010/07/19/ab-solve-beginners-guide-to-entity-framework.aspx>
- .NET 4.0-Update (Deutsch)
ISBN 3866454430
- Programming Entity Framework
ISBN 0596807260

Kontakt und weitere Infos

- **Manfred Steyer**
- www.softwarearchitekt.at
- manfred.steyer@softwarearchitekt.at