

.NET Entity Framework 4

Grundlagen und Einsatz in .NET 4.0

W3L AG
info@W3L.de

2010



Agenda

■ Motivation

- OR-Mapping-Frameworks

■ Entity Framework 4

- Entwicklung
- Mapping & Architektur
- Generatorverfahren
- Praxisbeispiel
- Erweiterbarkeit
- Ausblick & Bewertung

■ Kurzfassung: NHibernate

■ EF4 vs. NHibernate



Objektrelationale Abbildung

■ Was ist OR-Mapping?

- Technik zur Persistierung von Objekten in eine relationale Datenbank
- Objekte und ihre Beziehungen werden in einem relationalen Datenbankschema abgebildet
 - Klassen werden zu Relationen (Attribute = Spalten)
 - Ein Objekt wird zu einem Datensatz
 - Assoziationen werden über Fremdschlüsselbeziehungen abgebildet
- Verlust der Objektidentität
 - Einsatz von Primärschlüsselattributen zur Identifikation der Datensätze

OR-Mapping: Probleme

■ Probleme

- Die Implementierung eines OR-Mappings ist zeitaufwendig
 - Schritte:
 - Datenbankschema entwickeln und implementieren
 - DB-Schnittstellentreiber einbinden (ODBC/JDBC oder native Treiber)
 - Verbindungsmanagement implementieren (Connection Pooling etc.)
 - Mapper-Methoden implementieren (Objekt \leftrightarrow Datentupel)
- Verschiedene SQL-Dialekte und DB-spezifische Funktionen müssen berücksichtigt werden

■ Lösung: „ORM-Frameworks“

ORM-Frameworks

■ ORM-Frameworks

- Ermöglichen es, verschiedene Objekte in relationalen Datenbanken zu speichern
- Datenbankschema wird durch Metainformationen beschrieben
- Abstraktion von der Datenbank
 - Unabhängige Abfragemöglichkeiten z. B.: HQL oder Entity SQL
 - Problemloser Austausch des Datenbanksystems
- Verbessern die Performanz durch Lazy-Loading und Caching
- Vorgehensweise:
 - Ein Modell beschreibt die Umsetzung des Fachkonzepts in ein relationales Datenbankschema (Häufig durch ein XML-Dokument)

■ ORM-Frameworks in .NET:

- Entity Framework
- NHibernate
- Telerik OpenAccess ORM (ehem. Vanatec)

ADO.NET Entity Framework

■ Was ist das Entity Framework?

- Konzept für die objektorientierte Modellierung von persistenten Objekten
- Gehört zu den ADO.NET-Komponenten im .NET-Framework
- Die Modellierung erfolgt mit dem „ADO.NET Entity Data Model Designer“
 - Bestandteil von Visual Studio ab 2008 SP1
- Das Modell wird als „Entity Data Model“ bezeichnet (XML-Dokument)

■ Entwicklung

- LINQ-to-SQL – .NET ab Version 3.5 (nur MS SQL Server)
- Entity Framework 1 – .NET ab Version 3.5 SP1
- Entity Framework 2 – Herausgabe mit Visual Studio 2010 und .NET 4
 - Wird als „Entity Framework 4“ bezeichnet (EF4) ← → .NET 4

Entity Framework

■ Verfahren:

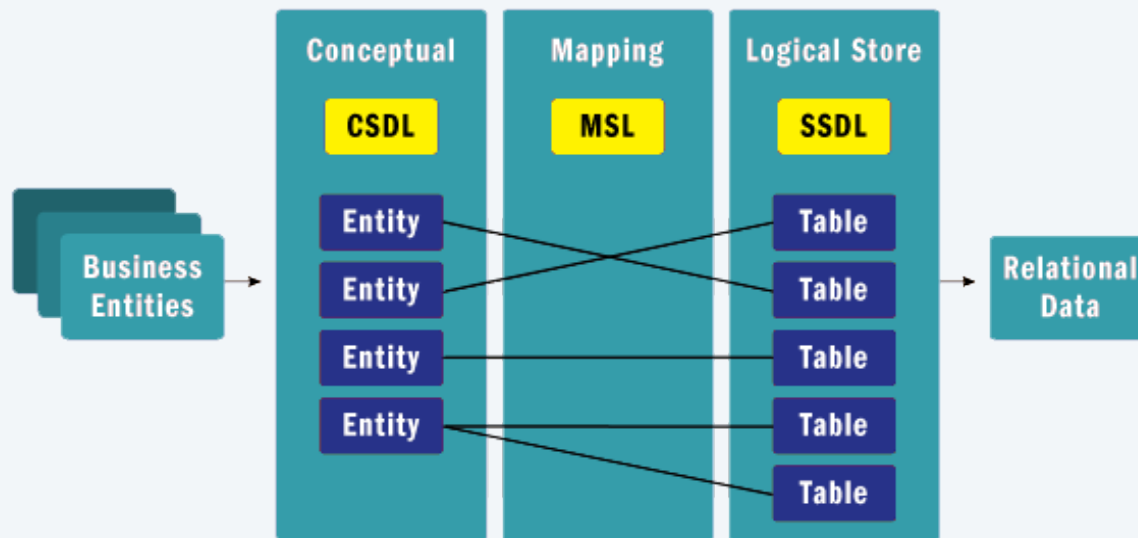
- Ein „Entity Data Model“ beschreibt das Fachkonzept und dessen objektrelationale Abbildung
- Vorgehensweisen:
 - Reverse Engineering – Das Entity Data Model wird aus einem bestehenden Datenbankschema generiert (EF1 – Database First)
 - Forward Engineering – Aus dem EDM wird die Fachklasse und das Datenbankschema generiert (EF4 – Model First)

■ Datenbankbindung

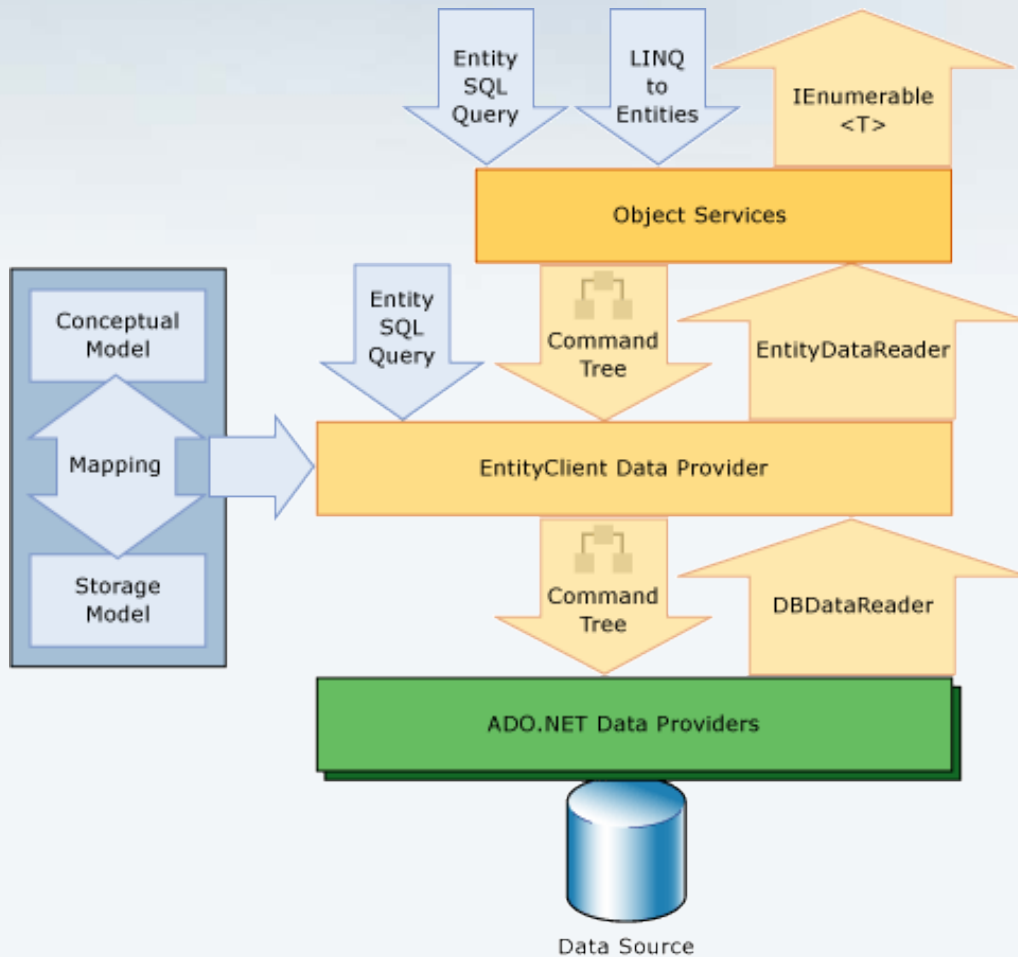
- Über einen „ADO.NET Data Provider“ wird die Datenbank angesprochen
 - Dieser Provider muss das EF unterstützen
 - Vorhanden für:
 - Microsoft SQL und MySQL
 - Oracle: ab Anfang 2011
(<http://www.oracle.com/technetwork/topics/dotnet/whatsnew/oracle-entity-framework-sod-130214.pdf>)

Entity Framework: Mapping

- Das Mapping besteht aus 3 Stufen → „Entity Data Model“ (EDM)
 - CSDL: Conceptual Schema Definition Language (Objekt-Modell)
 - MSL: Mapping Schema Language (Zuordnung zwischen Objekt und Datensatz)
 - SSDL: Store Schema Definition Language (Datenbankschema)

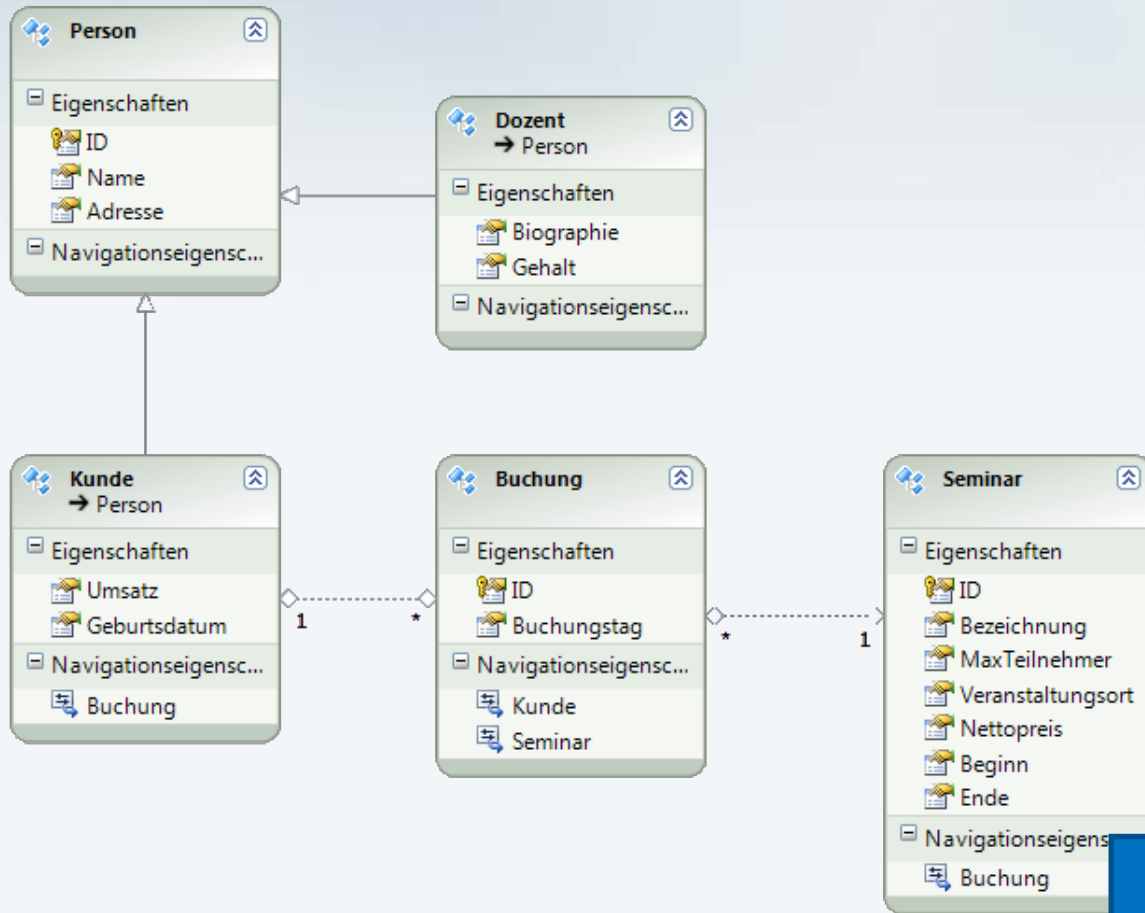


Entity Framework: Architektur



- Umwandlung in Objekte
- Liefert einen Stream von Zeilen
- Kapselt den Data Provider
- Umwandlung in SQL-Statements
- Liefert einen Stream von Zeilen
- Datenbankschnittstelle
- MS SQL- oder MySQL-Datenbank

Praxisbeispiel



Quellcode-Generierung

■ Text Template Transformation Toolkit (T4)

- Ein Verfahren für die Codegenerierung
- Eine Transformationsvorlage beschreibt das Vorgehen (.tt-Datei)
- Bestandteil von Visual Studio ab 2008
 - Allerdings ohne eigenen T4-Editor (kein Intelli-Sense & Syntax-Highlighting)
 - Plug-In: tangible T4-Editor (<http://t4-editor.tangible-engineering.com>)
- Einsatz im Entity-Framework:
 - Über eine T4-Vorlage werden aus dem „Entity Data Model“ die erforderlichen Klassen generiert
 - Microsoft stellt zwei Vorlagen:
 - ADO.NET EntityObject Generator
 - Alle Entites erben von EntityObject (+ Änderungsverfolgung)
 - ADO.NET POCO Entity Generator (Onlinevorlage)
 - Plain Old CLR Object (schmalere Objekte - WCF)

Performanz-Mechanismen

■ Eager Loading (mit Include)

- Objekte mit Abhängigkeiten laden
- Include:
 - `context.Customers.Include("Orders");`
 - Kunden direkt mit ihren Bestellungen laden

■ Lazy-Loading

- EF1: Objekte werden nicht automatisch nachgeladen (Explizit mit `load()`)
 - `if(!oneCustomer.Orders.IsLoaded) oneCustomer.Orders.Load();`
- EF4: Lazy-Loading ist Transparent
 - Kein Aufruf der `Load()`-Methode mehr nötig

■ Caching

- Der „ObjectContext“ stellt einen Container für Objekte im Arbeitsspeicher dar
 - Mit Identitätsauflösung, Zustandsverwaltung und Änderungsnachverfolgung
- Ein Second-Level-Cache wird nicht angeboten

Erweiterbarkeit

■ Erweitern von generierten Klassen

- Zusätzliche Methoden in partiellen Klassen implementieren
- Erweiterung durch Extension Methods (statische Funktionen)

■ Code-Generator

- T4-Vorlage modifizieren oder eigene Vorlage schreiben
- Einflussnahme auf Standard-Operationen wie z. B. DeleteObject() möglich

■ Eigenen Entity-Framework-Provider schreiben (Aufwendig!!)

- Second-Level-Cache
- Protokollierung implementieren
- Beispiel:
 - <http://code.msdn.microsoft.com/EFSampleProvider>
 - <http://code.msdn.microsoft.com/EFProviderWrappers>

Ausblick

■ Vorgehensweise: Code First Programming

■ Fachklassen als POCOs selbst implementieren

■ Mapping:

Direkt über Attribute in den Klassen (Java: Annotations)

- [Key]
public int ISBN { get; set; }
- [Required]
[MaxLength(120)]
public string Title { get; set; }

Als Quellcode im „DbContext“ (OnModelCreating)

- builder.Entity<Book>().HasKey(b => b.ISBN);
- builder.Entity<Book>().Property(b => b.Title).IsRequired();

■ Bereits in der „Community Technology Preview 4“ verfügbar

<http://tinyurl.com/EF4CTP4>

Entity Framework 4

Vorteile:

- GUI-Unterstützung durch Visual Studio 2010
- Rapid Development
- Support durch Microsoft
- Generiert Fachklassen und Datenbankschema
- LINQ-Unterstützung
- Datenbindung durch EntityDataSource-Objekte

Nachteile:

- Modell kann nicht zur Laufzeit verändert werden
- Modell basiert auf einem sehr komplexen XML-Schema
- Wenige Data-Provider vorhanden
- Kein 2nd Level Cache
- „Noch“ ein junges Framework
- Generierte Fachkonzept-Klassen dürfen nicht modifiziert werden
 - Erweiterung durch „partielle Klassen“

Kurzfassung

NHIBERNATE

NHibernate

■ NHibernate

- Portierung von Hibernate nach .NET
- Wird seit Dezember 2001 als quelloffenes Projekt entwickelt
- Unterstützung von RedHat (JBoss-Community)
- Verfahren:
 - Ein XML-Dokument beschreibt die Umsetzung eines Fachkonzepts auf ein relationales Datenbankschema
 - Das Datenbankschema kann aus dem Modell generiert werden
 - Verwendet eine „ADO.NET Connection“ für die Kommunikation mit der Datenbank
 - Der Data Provider muss den Funktionsumfang von .NET 2.0 unterstützen
 - Vorhanden für:
 - MS-SQL; Oracle; MySQL; SQLite; PostgreSQL u. FireBird

NHibernate

Vorteile:

- Basiert auf Hibernate
- Offener Quellcode → Erweiterbar
- Klares Mapping (XML)
- Unterstützung von SQLite als dateibasierte Datenbank
- DB-Modell kann zur Laufzeit geändert/erweitert werden
- Data-Provider für alle relevanten Datenbankserver

Nachteile:

- Hohe Komplexität durch den großen Funktionsumfang
- Keine GUI-Unterstützung
- Fachklassen müssen selbst implementiert werden
- LINQ-Unterstützung erst ab Version 3.x

Der Vergleich

Entity Framework 4

vs.

NHibernate

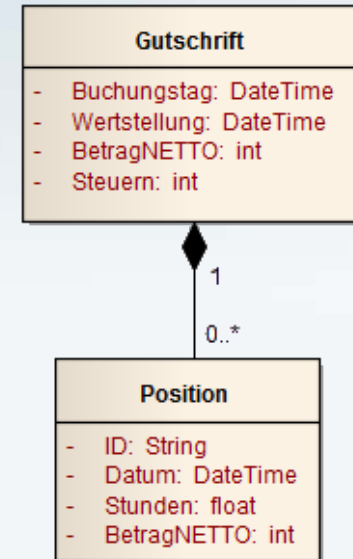
Benchmark

■ Bedingungen:

- Entity Framework 4 (4.0.0.0 [NET4])
 - Lazy-Loading: Aktiv
 - POCO-Vorlage
- NHibernate (3.0.0.2002 [Beta2])
 - Lazy-Loading: Aktiv (Castle-Provider)
 - Second-Level-Cache: Deaktiviert
- Datenbank (Lokal)
 - Microsoft SQL2008 - Version: 10.50.1600.1

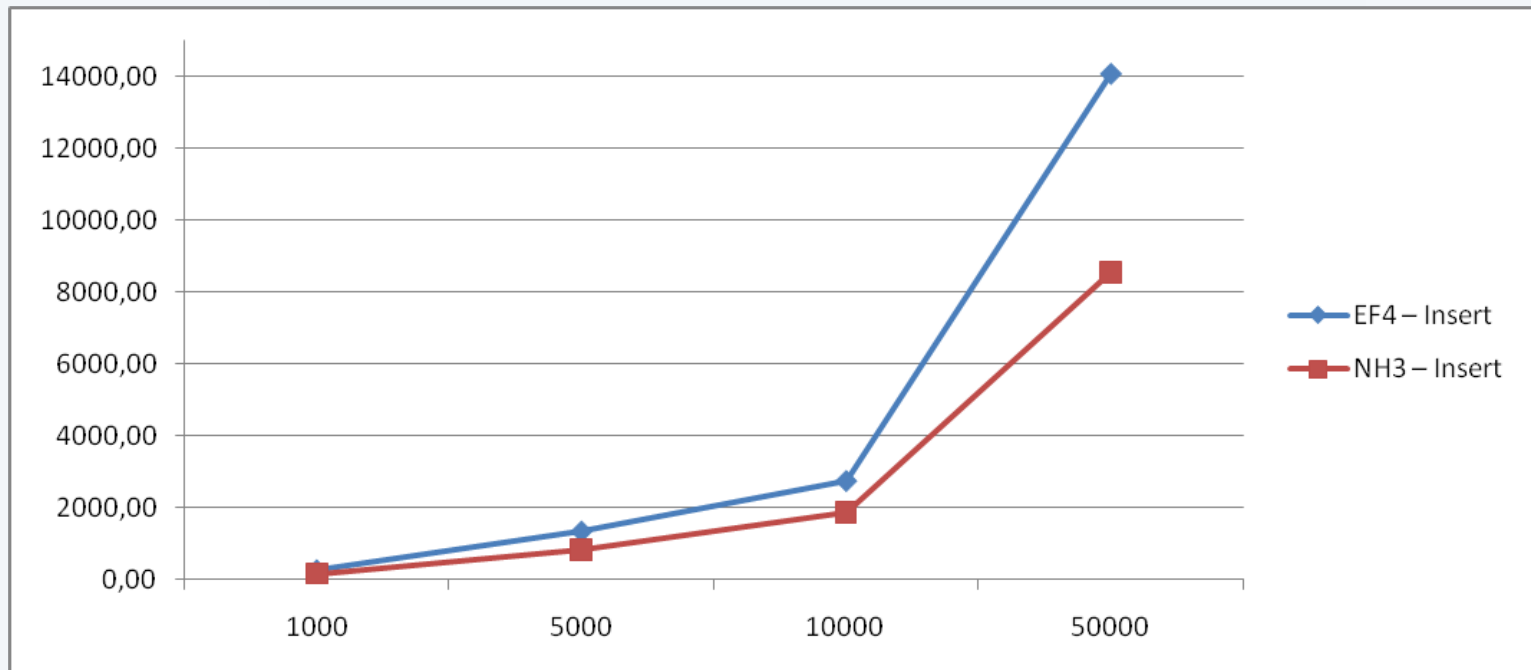
■ Messungen:

- Insert: Datensatz anlegen (Gutschrift mit Position)
- Select: Positionen selektieren (LINQ) und Iteration über die Ergebnismenge
- Update: Laden und Ändern einer Position



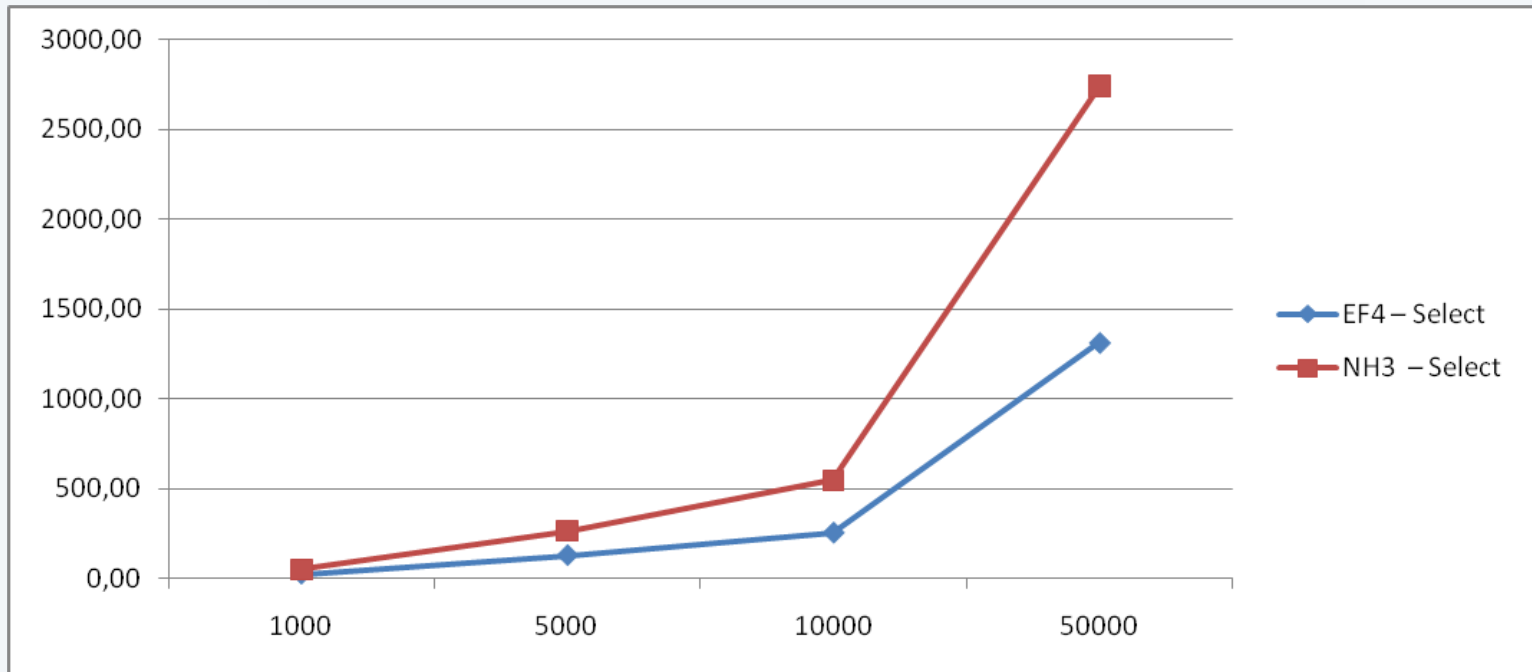
Benchmark – Insert

| Operation | 1.000 | 5.000 | 10.000 | 50.000 |
|--------------|------------|-------------|-------------|--------------|
| EF4 – Insert | 267,8 [ms] | 1346,7 [ms] | 2751,3 [ms] | 14085,2 [ms] |
| NH3 – Insert | 201,8 [ms] | 876,3 [ms] | 1873,7 [ms] | 8852,8 [ms] |



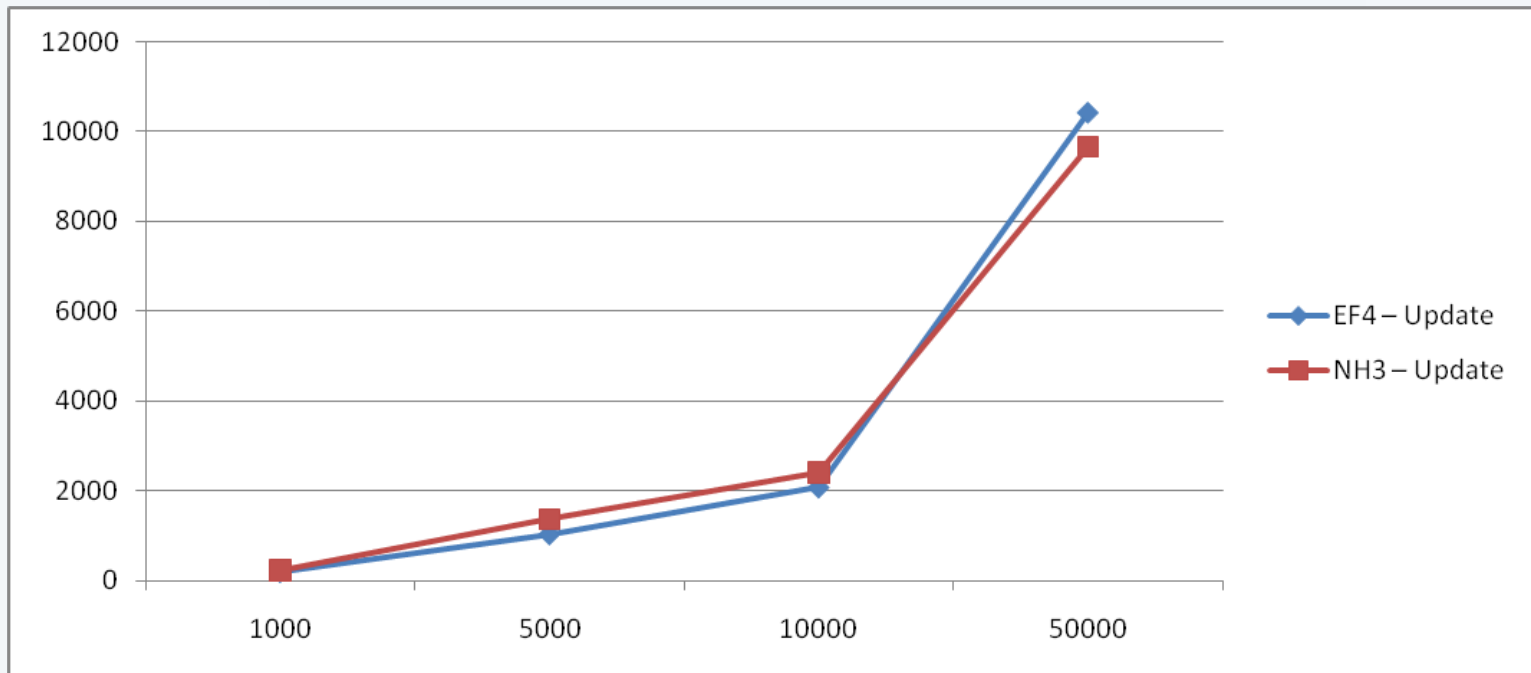
Benchmark – Select

| Operation | 1.000 | 5.000 | 10.000 | 50.000 |
|--------------|-----------|------------|------------|-------------|
| EF4 – Select | 24,2 [ms] | 127,8 [ms] | 253,6 [ms] | 1314,4 [ms] |
| NH3 – Select | 50,7 [ms] | 264,5 [ms] | 546,5 [ms] | 2742,1 [ms] |



Benchmark – Update

| Operation | 1.000 | 5.000 | 10.000 | 50.000 |
|--------------|------------|-------------|-------------|--------------|
| EF4 – Update | 185,7 [ms] | 1028,1 [ms] | 2074,9 [ms] | 10430,7 [ms] |
| NH3 – Update | 229,4 [ms] | 1371 [ms] | 2405,8 [ms] | 9661,7 [ms] |



Welcher Mapper ist der Richtige?

NHibernate

Volle Kontrolle über
das OR-Mapping

Besser bei komplexen
Datenbankstrukturen

Enterprise-Projekte

Entity Framework

Unterstützung durch
Visual Studio (GUI)

„Rapid Development“ –
schnelle Ergebnisse

Kleine Projekte mit
geringerer Komplexität

Vielen Dank!

Inhouse-Schulungen



Wir bieten Inhouse-Schulungen und Beratung durch unsere IT-Experten und -Berater.

Schulungsthemen

- Softwarearchitektur (OOD)
- Requirements Engineering (OOA)
- Nebenläufige & verteilte Programmierung

Gerne konzipieren wir auch eine individuelle Schulung zu Ihren Fragestellungen.



Sprechen Sie uns an!
Tel. 0231/61 804-0, info@W3L.de

W3L-Akademie



Flexibel online lernen und studieren!

In Zusammenarbeit mit der Fachhochschule Dortmund bieten wir

zwei Online-Studiengänge

- B.Sc. Web- und Medieninformatik
- B.Sc. Wirtschaftsinformatik

und 7 Weiterbildungen im IT-Bereich an.



Besuchen Sie unsere Akademie!
<http://Akademie.W3L.de>