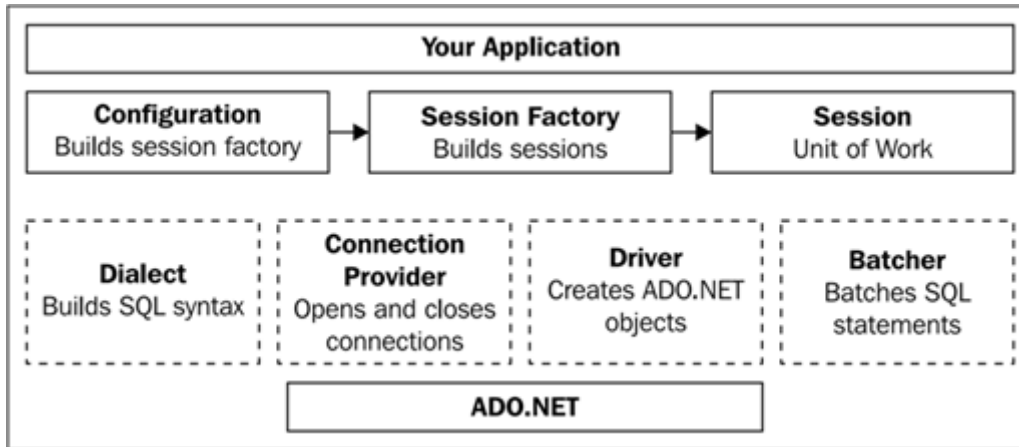


# Chapter 1: Configure and Schema



Browse Installed Updates NuGet Package Manager: Eg.Core

NHibernate Include prerelease Package source: nuget.org

**NHibernate** by NHibernate community, Hibernate co v4.0.4.4000  
NHibernate is a mature, open source object-relational mapper for the .NET framework. It is actively developed...

**FluentNHibernate** by James Gregory and contributors, 8 v2.0.3  
Fluent, XML-less, compile safe, automated, convention-based mappings for NHibernate.

**NHibernate.Envers** by NHibernate community, Hibernate v2.1.0  
Audit trail integrated with NHibernate.

**NHibernate.Mapping.Attributes** by Pierre Henri Ku v4.0.0.1  
Attribute based mapping for NHibernate.

**NHibernate.Search** by NHibernate Community, 5,1 v2.0.2.4000  
NHibernate Search brings the power of full text search engines to the persistence domain model by combin...

**Abp.NHibernate** by Abp.NHibernate, 27,2K download v0.10.3.1  
MongoDB integration package for ASP.NET Boilerplate.

**NHibernate.Search.MB** by NHibernate Community v4.0.4.4000  
NHibernate Search build against NHibernate 4.0.4.4000 and Lucene 3.0.3

**NHibernate**

Version: Latest stable 4.0.4.4000 Install

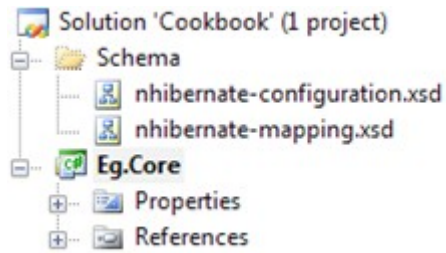
Options

**Description**  
NHibernate is a mature, open source object-relational mapper for the .NET framework. It is actively developed, fully featured and used in thousands of successful projects.

**Version:** 4.0.4.4000  
**Author(s):** NHibernate community, Hibernate community  
**License:** <https://raw.githubusercontent.com/nhibernate/nhibernate-core/master/lgpl.txt>  
**Date published:** Monday, August 17, 2015 (8/17/2015)  
**Project URL:** <http://nhibernate.info>  
**Report Abuse:** <https://www.nuget.org/packages/NHibernate/4.0.4.4000/ReportAbuse>  
**Tags:** DataBase, DAL, ORM, ObjectRelationalMapping

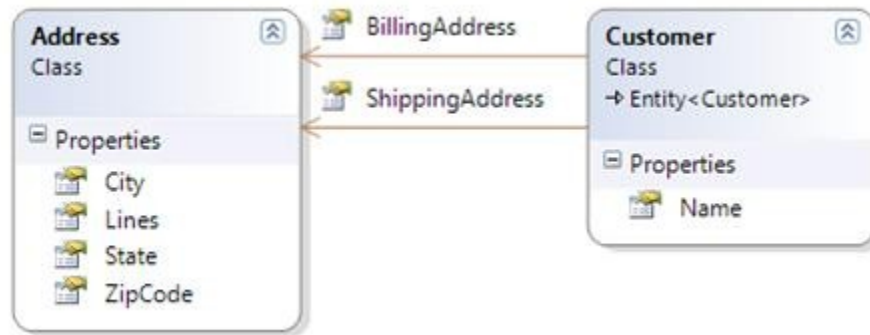
**Dependencies**  
Iesi.Collections (>= 4.0.0 && < 5.0.0)

## Chapter 2: Models and Mappings



Product			
	Column Name	Data Type	Allow Nulls
🔑	Id	uniqueidentifier	<input type="checkbox"/>
	Name	nvarchar(255)	<input type="checkbox"/>
	Description	nvarchar(255)	<input checked="" type="checkbox"/>
	UnitPrice	decimal(19, 5)	<input type="checkbox"/>
			<input type="checkbox"/>

Product			
	Column Name	Data Type	Allow Nulls
🔑	Id	uniqueidentifier	<input type="checkbox"/>
	ProductType	nvarchar(255)	<input type="checkbox"/>
	Name	nvarchar(255)	<input type="checkbox"/>
	Description	nvarchar(255)	<input checked="" type="checkbox"/>
	UnitPrice	decimal(19, 5)	<input type="checkbox"/>
	Director	nvarchar(255)	<input checked="" type="checkbox"/>
	Author	nvarchar(255)	<input checked="" type="checkbox"/>
	ISBN	nvarchar(255)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>



ActorRole			
	Column Name	Data Type	Allow Nulls
🔑	Id	uniqueidentifier	<input type="checkbox"/>
	Actor	nvarchar(255)	<input type="checkbox"/>
	Role	nvarchar(255)	<input type="checkbox"/>
	MovieId	uniqueidentifier	<input checked="" type="checkbox"/>
	ActorIndex	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

```

INSERT INTO Product (Name, Description, UnitPrice, Director, ProductType, Id) VALUES (@p0, @p1, @p2, @p3, 'Eg.Core.Movie', @p4);
INSERT INTO ActorRole (Actor, Role, Id) VALUES (@p0, @p1, @p2);
UPDATE ActorRole SET MovieId = @p0, ActorIndex = @p1 WHERE Id = @p2;
  
```

```

INSERT INTO Student (Name, Id) VALUES (@p0, @p1);
INSERT INTO Student (Name, Id) VALUES (@p0, @p1);
INSERT INTO Course (Name, Id) VALUES (@p0, @p1);
INSERT INTO Course (Name, Id) VALUES (@p0, @p1);
INSERT INTO CourseStudent (CourseId, StudentId) VALUES (@p0, @p1);
INSERT INTO CourseStudent (CourseId, StudentId) VALUES (@p0, @p1);
INSERT INTO CourseStudent (CourseId, StudentId) VALUES (@p0, @p1);
SELECT course0_Id as Id0_0_, course0_Name as Name0_0_ FROM Course course0_ WHERE course0_Id=@p0;
SELECT students0_CourseId as CourseId1_, students0_StudentId as StudentId1_, student1_Id as Id2_0_, student1_Name as Name2_0_ FROM
  
```

```

INSERT INTO Customer (Name, Id) VALUES (@p0, @p1);
INSERT INTO CustomerAddress (CustomerId, AddressIndex, AddressLine1, AddressLine2, City, PostalCode) VALUES (@p0, @p1, @p2, @p3, @p4, @p5);
INSERT INTO CustomerAddress (CustomerId, AddressIndex, AddressLine1, AddressLine2, City, PostalCode) VALUES (@p0, @p1, @p2, @p3, @p4, @p5);
INSERT INTO CustomerTag (CustomerId, Tag) VALUES (@p0, @p1);
INSERT INTO CustomerTag (CustomerId, Tag) VALUES (@p0, @p1);
  
```

```
Commit 1
NHibernate.StaleObjectStateException: Row was updated or deleted by another transaction (or unsaved-value mapping was incorrect): [MappingRecipes.Versioning.VersionedProduct#1]
   at NHibernate.Persister.Entity.AbstractEntityPersister.Update(Object id, Object[] fields, Object[] oldFields, Object rowId, Boolean[] includeProperty, Int32 j, Object oldVersion, Object obj, SqlCommandInfo sql, ISessionImplementor session)
   at NHibernate.Persister.Entity.AbstractEntityPersister.UpdateOrInsert(Object id, Object[] fields, Object[] oldFields, Object rowId, Boolean[] includeProperty, Int32 j, Object oldVersion, Object obj, SqlCommandInfo sql, ISessionImplementor session)
```

```
INSERT INTO Project (Id) VALUES (@p0);
INSERT INTO Project (Id) VALUES (@p0);
INSERT INTO "Order" (Id) VALUES (@p0);
INSERT INTO OrderItem (Name, OrderId, Id) VALUES (@p0, @p1, @p2);
INSERT INTO OrderItem (Name, OrderId, Id) VALUES (@p0, @p1, @p2);
INSERT INTO "Order" (Id) VALUES (@p0);
INSERT INTO OrderItem (Name, OrderId, Id) VALUES (@p0, @p1, @p2);
INSERT INTO OrderItem (Name, OrderId, Id) VALUES (@p0, @p1, @p2);
INSERT INTO OrderProject (ProjectId, OrderId) VALUES (@p0, @p1);
INSERT INTO OrderProject (ProjectId, OrderId) VALUES (@p0, @p1);
INSERT INTO OrderProject (ProjectId, OrderId) VALUES (@p0, @p1);
SELECT this_Id as Id3_0_ FROM Project this_
SELECT orders0_ProjectId as ProjectId1_, orders0_OrderId as OrderId1_, order1_Id as Id1_0_ FROM OrderProject orders0_ left outer join
SELECT items0_OrderId as OrderId1_, items0_Id as Id1_, items0_Id as Id0_0_, items0_Name as Name0_0_, items0_OrderId as OrderId0,
SELECT items0_OrderId as OrderId1_, items0_Id as Id1_, items0_Id as Id0_0_, items0_Name as Name0_0_, items0_OrderId as OrderId0,
SELECT orders0_ProjectId as ProjectId1_, orders0_OrderId as OrderId1_, order1_Id as Id1_0_ FROM OrderProject orders0_ left outer join
```

## Chapter 3: Sessions and Transactions

```
Building session provider
Building session provider
Opening session
All Books
 * NHibernate 3.0 Cookbook
Opening session
All Movies
 * Raiders of the Lost Ark
 * The Bucket List
Disposing session
Disposing session
Building session provider
Opening session
All Products
 * Raiders of the Lost Ark
 * The Bucket List
 * NHibernate 3.0 Cookbook
Disposing session
Press any key
_
```

[Application name](#)   [Home](#)   [About](#)   [Contact](#)

### Books

- NHibernate 3.0 Cookbook - Jason Dentler

```
PerformSave
Id:4
PerformUpdate
Saving TestPublisherAuto
Searching for TestPublisherAuto
Found it!
Searching for TestPublisherAuto again
Found it!
Saving TestPublisherCommit
Searching for TestPublisherCommit
Didn't find it!
Searching for TestPublisherCommit again
Found it!
Saving TestPublisherNever
Searching for TestPublisherNever
Didn't find it!
Searching for TestPublisherNever again
Didn't find it!
```

Executing with None and None

```
Thread1 acquiring lock
Thread1 acquired lock
Thread2 acquiring lock
Thread2 acquired lock
Thread2 committed
Thread1 committed
```

Executing with Upgrade and Upgrade

```
Thread1 acquiring lock
Thread1 acquired lock
Thread2 acquiring lock
Thread1 committed
Thread2 acquired lock
Thread2 committed
```

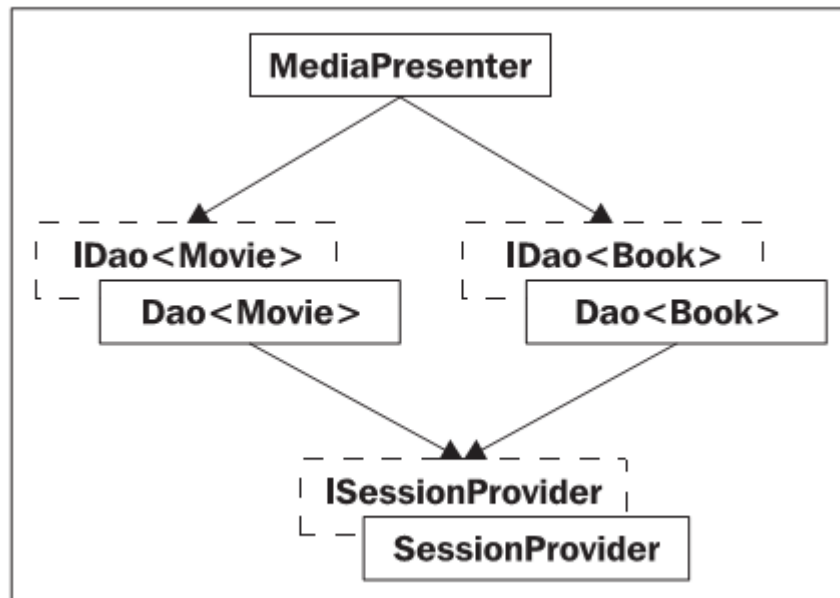
Executing with UpgradeNowait and UpgradeNowait

```
Thread1 acquiring lock
Thread1 acquired lock
Thread2 acquiring lock
Exception in Thread2:could not lock: [NH4CookbookHelpers.Queries.Model.Movie#1][SQL: SELECT Id FROM Product
with (updlock, rowlock, nowait) WHERE Id = ?]
Thread1 committed
```

```
Adding Sprockets.  
Adding Sprockets to product catalog.  
Adding Sprockets to warehouse system.  
Updating Sprockets.  
Updating Sprockets in product catalog.  
Updating Sprockets in warehouse system.  
Removing Sprockets.  
Removing Sprockets from product cata-log.  
Removing Sprockets from warehouse system.  
Product could not be removed.  
Warehouse still has inventory of Sprockets.
```

```
Saving product in session A.  
Changing price in session B.  
Price was 10,00 kr  
Refreshing  
Price is 15,00 kr
```

```
Saving product in session A.  
Changing price in session B.  
Price was 10,00 kr  
Refreshing  
Price is 15,00 kr
```



## Chapter 4: Queries

```
Movies directed by Spielberg:
9,59 kr Raiders of the Lost Ark starring Harrison Ford & Karen Allen

Movies with Morgan Freeman:
15,00 kr The Bucket List starring Jack Nicholson & Morgan Freeman

This book:
50,00 kr NHibernate 3.0 Cookbook - Packt Publishing (ISBN 978-1-849513-04-3)

Cheap products:
9,59 kr Raiders of the Lost Ark starring Harrison Ford & Karen Allen
15,00 kr The Bucket List starring Jack Nicholson & Morgan Freeman
```

```
Movie Price List:
9,59 kr Raiders of the Lost Ark
15,00 kr The Bucket List

Average Movie Price:
12,30 kr

Average Price by Director:
15,00 kr Rob Reiner
9,59 kr Steven Spielberg
```



Movies directed by Spielberg:

9,59 kr Raiders of the Lost Ark starring Harrison Ford & Karen Allen

Movies with Morgan Freeman:

15,00 kr The Bucket List starring Jack Nicholson & Morgan Freeman

This book:

50,00 kr NHibernate 3.0 Cookbook - Packt Publishing (ISBN 978-1-849513-04-3)

Cheap products:

9,59 kr Raiders of the Lost Ark starring Harrison Ford & Karen Allen

15,00 kr The Bucket List starring Jack Nicholson & Morgan Freeman

Movie Price List:

9,59 kr Raiders of the Lost Ark

15,00 kr The Bucket List

Average Movie Price:

12,30 kr

Average Price by Director:

15,00 kr Rob Reiner

9,59 kr Steven Spielberg

# Chapter 6: Testing

The screenshot displays the NHibernate Profiler application interface. At the top, the title bar reads "NHibernate Profiler" with menu options: FILE, OPTIONS, REPORTS, HELP. Below the title bar, a status bar indicates "Recording 32 days remaining. Build: 3088." and "Filter Inactive".

The main interface is divided into several panels:

- Sessions:** Shows "Recent Statements" and "Session #1 - 0.852 [1]".
- Statements:** A table with columns "Short SQL", "Row Count", "Duration", and "Alerts". It contains one entry: "SELECT ... FROM Product book0\_ WHERE book0\_ProductType =..." with a row count of 1 and a duration of 5 ms / 43 ms.
- Details:** Shows the full SQL query and its results. The query is:

```
1 select book0_.Id as Id0_,
2 book0_.Version as Version0_,
3 book0_.Name as Name0_,
4 book0_.Description as Descript5_0_,
5 book0_.UnitPrice as UnitPrice0_,
6 book0_.Author as Author0_,
7 book0_.ISBN as ISBN0_
8 from Product book0_
9 where book0_.ProductType = 'Eg.Core.Book'
10 and book0_.Author = 'Jason Dentler' /* @p0 */
```

The results table has columns "Name", "Value", and "Type". One row is shown: "@p0", "Jason Dentler", "String (4000)".
- Session factory Statistics:** Lists various statistics for the "unnamed" session factory, such as "Close Statement Count", "Collection Fetch Count", etc., all with a value of 0, except for "Connect Count" which is 1.

At the bottom of the Details panel, it says "Started at 09:45:08.196 PM See the 1 row(s) resulting from this statement. Query plan for this statement."

# NHibernate Profiler

FILE OPTIONS REPORTS HELP [Icons]

Recording 32 days remaining. Build: 3088. Filter Inactive

**Sessions** Analysis

Recent Statements

- ☆ Session #1 - 0.852 [1]

**Session #1**

Statements Entities Session Usage

Short SQL	Row Count	Duration	Alerts
☆ SELECT ... FROM Product book0_ WHERE book0_ProductType =...	1	5 ms / 43 ms	[Alert Icon]

Details 1 Alert Stack Trace 1 Table

```

1 select book0_.Id as Id0_,
2 book0_.Version as Version0_,
3 book0_.Name as Name0_,
4 book0_.Description as Descript5_0_,
5 book0_.UnitPrice as UnitPrice0_,
6 book0_.Author as Author0_,
7 book0_.ISBN as ISBN0_
8 from Product book0_
9 where book0_.ProductType = 'Eg.Core.Book'
10 and book0_.Author = 'Jason Dentler' /* @p0 */

```

Name	Value	Type
@p0	'Jason Dentler'	String (4000)

Started at 09:45:08.196 PM See the 1 row(s) resulting from this statement. Query plan for this statement.

**Session factory Statistics**

unnamed

- Close Statement Count 0
- Collection Fetch Count 0
- Collection Load Count 0
- Collection Recreate Count 0
- Collection Remove Count 0
- Collection Role Names Eg.Core.Mov
- Collection Update Count 0
- Connect Count 1

# NHibernate Profiler

FILE OPTIONS REPORTS HELP [Icons]

Recording 32 days remaining. Build: 3088. Filter Inactive

**Sessions** Analysis

Recent Statements

- ☆ Session #1 - 0.852 [1]

**Session #1**

Statements Entities Session Usage

Short SQL	Row Count	Duration	Alerts
☆ SELECT ... FROM Product book0_ WHERE book0_ProductType =...	1	5 ms / 43 ms	[Alert Icon]

Details 1 Alert Stack Trace 1 Table

```

Remotion.Linq.Remotion.Linq.QueryableBase`1.GetEnumerator
In 0 | col 0 | Source file could not be found.
NHProfTest.NHProfTest.Program.Main
In 34 | Program.cs | Double-click to jump to the code.
col 32 |
System.System.AppDomain_nExecuteAssembly
In 0 | | Source file could not be found.
col 0 |
System.System.AppDomain.ExecuteAssembly
In 0 | | Source file could not be found.
col 0 |

```

**Session factory Statistics**

unnamed

- Close Statement Count 0
- Collection Fetch Count 0
- Collection Load Count 0
- Collection Recreate Count 0
- Collection Remove Count 0
- Collection Role Names Eg.Core.Mov
- Collection Update Count 0
- Connect Count 1

Chrome 52.0 http://localhost:49349/Books

Cache Configuration Environment Execution Metadata Model Binding **NHibernate** Request Routes Server Session Ajax History

Section	Content		
Request Summary	<b>Selects</b>	<b>Inserts</b>	<b>Updates</b>
	1	0	0
Request Details	<i>Log</i>		
	<pre>6bfa2b9d-0b41-4440-9a78-145f6220457d SELECT this_.Id as Id0_0_, this_.Name as Name0_0_, this_.Description as Descript4_0_0_, this_.UnitPrice as UnitPrice0_0_, this_.ISBN as ISBN0_0_0_</pre>		
	<pre>[ 'Stack Trace' : 'ActionFilter...', 'Stack Trace' : 'ActionFilter...', length=49 ]</pre>		
	Command: ExecuteReader took 0 ms		
	Command: DataReader was closed after 0 ms		

## Chapter 9: NHibernate Contribution Projects

Database Configuration

Server name:

Log on to the server

Use Windows Authentication

Use SQL Server Authentication

User name:

Password:

Save my password

Connect to a database

Select or enter a database name:

Attach a database file:

Logical name:

