# NH常用默认配置：

name对应model中的Property,column 对应table中的列名；

默认column name等于property name;

mappping files ends with ".hbm.xml";

property type 可以省略，可以从Model的类型中推断；

not-null="false";

db中table name默认与model的class name一样。

natural id: unique,not null

所有public 属性，方法都要标记为virtual

# Inverse ,cascade

Inverse 用户设置类之间的关联。谁来管理这个关联（Foreign key）.

在one-to-many 中设置在many端。

<class name=”Parent”>

<bag name=”SubItems” inverse=”false”>

<key column=”PK\_ID”>

<one-to-many class=”SubItem”/>

</bag>

</class>

**Inverse=false:**

表示parent有责任保存与subItem之间的关联信息。当执行session.Save(parent);时候，会自动保存SubItem的PK\_ID;

对应SQL按 “Insert/Update ” strategy来执行，即先Insert parent 和subItems (PK\_ID=null)，在update subItems的PK\_ID (parentId);

Bellow:Inverse=false(default),cascade=”none”(default)

var category = new Category { Name = "Category1" };

var product1 = new Product { Name = "Product1", Discontinued = false };

var product2 = new Product { Name = "Product2", Discontinued = false };

var product3 = new Product { Name = "Product3", Discontinued = false };

// Associate the products with the category

category.Products = new List<Product> { product1, product2, product3 };

// Save everything in the session and commit the transaction

using (var session = this.sessionFactory.OpenSession())

using (var transaction = session.BeginTransaction())

{

session.Save(category);

*session.Save(product1);*

*session.Save(product2);*

*session.Save(product3);*

transaction.Commit();

}

// Get the values from the DB and check the results

using (var session = this.sessionFactory.OpenSession())

{

var categoryFromDb = session.Get<Category>(category.Id);

Assert.IsNotNull(categoryFromDb.Products);

Assert.IsTrue(categoryFromDb.Products.Count == 3);

}

|  |  |
| --- | --- |
| Inverse=false | means that when you save the Category you will also save the association of each Product that is inside the **Category.Products** collection |
| Inverse=true | the Parent does not have the responsibility of saving the association |

**Inverse** can be used only to save entity association info

It tells NHibernate that the Parent is responsible (or not) of saving the **association** to its childs

这里还是需要自己保存prod1,prod2,prod3,因为Inverse设置谁来保存关联信息，不保存关联对应的实体（如果cascade=”none”，意味着需要自己保存关联entity，如果不同时保存关联实体，将会报TransientObjectException错误，因为持久化Category时候，Category.Products没有持久化，造成内存中数据与数据库中数据不一致（mismacth））

## **Cascade**

The **Cascade** mapping attribute helps NHibernate to decide which operations should be cascaded from the Parent object to the Child object