

با کمک Fluent NHibernate می‌توان نگاشت‌ها را به دو صورت [خودکار](#) و یا [دستی](#) تعریف کرد. در حالت خودکار، روابط بین کلاس‌ها بررسی شده و بدون نیاز به تعریف هیچگونه ویژگی (attribute) خاصی بر روی فیلدها، امکان تشخیص خودکار حالت‌های کلید خارجی، روابط یک به چند، چند به چند و امثال آن وجود دارد. یا اگر نیاز باشد تا اسکرپیت تولیدی جهت به روز رسانی بانک اطلاعاتی، طول خاصی را به فیلدی اعمال کند می‌توان از ویژگی‌های [NHibernate validator](#) استفاده کرد؛ مانند تعریف طول و نال نبودن یک فیلد که علاوه بر بکارگیری اطلاعات آن در حین تعیین اعتبار ورودی دریافتی، بر روی نحوه‌ی به روز رسانی بانک اطلاعاتی هم تاثیر گذار است:

```
public class Product
{
    public virtual int Id { set; get; }

    [Length(120)]
    [NotNullNotEmpty]
    public virtual string Name { get; set; }

    public virtual decimal UnitPrice { get; set; }
}
```

این نگاشت خودکار یا AutoMapping، تقریباً در 90 درصد موارد کافی است. فیلد Id را بر اساس یک سری پیش فرض‌هایی که این مورد هم قابل تنظیم است به صورت primary key تعریف می‌کند، طول فیلدها و نحوه‌ی پذیرفتن نال آن‌ها، از ویژگی‌های NHibernate validator گرفته می‌شود و روابط بین کلاس‌ها به صورت خودکار به روابط یک به چند و مانند آن ترجمه می‌شود و نیازی نیست تا کلاسی جداگانه را جهت مشخص سازی صریح این موارد تهیه کرد، یا ویژگی مشخص کننده‌ی دیگری را به فیلدها افزود. اما اگر برای مثال بخواهیم در این کلاس فیلد Name را به صورت Unique معرفی کنیم چه باید کرد؟ به عبارتی تمام آنچه که ویژگی AutoMapping در Fluent NHibernate انجام می‌دهد، بسیار هم عالی؛ اما فقط می‌خواهیم مقادیر یک فیلد منحصر بفرد باشد. برای این منظور اینترفیس [IAutoMappingOverride](#) تدارک دیده شده است:

```
public class ProductCustomMappings : IAutoMappingOverride<Product>
{
    public void Override(AutoMapping<Product> mapping)
    {
        mapping.Id(u => u.Id).GeneratedBy.Identity(); // ضروری است
        mapping.Map(p => p.Name).Unique();
    }
}
```

در حالت استفاده از اینترفیس IAutoMappingOverride مشخص سازی نحوه‌ی تولید primary key ضروری است و سپس برای نمونه، فیلد Name به صورت منحصر بفرد تعریف می‌گردد. در اینجا کل عملیات هنوز از روش AutoMapping پیروی می‌کند اما فیلد Name علاوه بر اعمال ویژگی‌های NHibernate validator، به صورت منحصر بفرد نیز معرفی خواهد شد.