

فرض کنید مدل کاربران سایت، دارای دو خاصیت راهبری (navigation properties) آدرس‌های مختلف یک کاربر و ایمیل‌های متفاوت او است:

```
public class SiteUser
{
    public int Id { get; set; }
    public string Name { get; set; }

    public virtual ICollection<Address> Addresses { get; set; }
    public virtual ICollection<Email> Emails { get; set; }
}

public class Email
{
    public int Id { get; set; }
    public string Text { get; set; }

    [ForeignKey("SiteUserId")]
    public virtual SiteUser SiteUser { get; set; }
    public int SiteUserId { get; set; }
}

public class Address
{
    public int Id { get; set; }
    public string Text { get; set; }

    [ForeignKey("SiteUserId")]
    public virtual SiteUser SiteUser { get; set; }
    public int SiteUserId { get; set; }
}
```

همچنین ViewModel ایی را هم که تعریف کرده‌ایم، شامل همان خواص راهبری مدل می‌شود:

```
public class UserViewModel
{
    public int Id { get; set; }
    public string Name { get; set; }

    public ICollection<Address> Addresses { get; set; }
    public ICollection<Email> Emails { get; set; }
}
```

در این حالت کوئری ذیل:

```
var user1 = context.Users.Project().To<UserViewModel>().FirstOrDefault();
```

سبب خواهد شد تا تمام خواص راهبری ذکر شده‌ی در ViewModel، در طی یک کوئری از بانک اطلاعاتی دریافت شده و مقدار دهی شوند. اما ... شاید در حین استفاده‌ی از آن، صرفاً به لیست ایمیل‌های شخص نیاز داشته باشیم و نیازی نباشد تا حتماً آدرس‌های او نیز واکشی شوند. برای حل این بارگذاری اضافی، می‌توان از تنظیم ExplicitExpansion استفاده کرد:

```
public class TestProfile : Profile
{
    protected override void Configure()
    {
        this.CreateMap<SiteUser, UserViewModel>()
            .ForMember(dest => dest.Addresses, opt => opt.ExplicitExpansion())
            .ForMember(dest => dest.Emails, opt => opt.ExplicitExpansion());
    }

    public override string ProfileName
    {

```

```
    get { return this.GetType().Name; }  
}
```

ExplicitExpansion به این معنا است که تا در کوئری مدنظر صریحا قید نشود که قرار است کدام خاصیت راهبری بسط یابد، اطلاعات آن از بانک اطلاعاتی دریافت نخواهد شد.

پس از تنظیم فوق، اگر کوئری ذکر شده را اجرا کنید، مشاهده خواهید کرد که دو خاصیت آدرس‌ها و ایمیل‌های شخص، نال هستند.

برای ذکر صریح خواص راهبری مورد نیاز، اینبار می‌توان از پارامترهای متد Project To مانند مثال ذیل استفاده کرد:

```
using (var context = new MyContext())  
{  
    var user1 = context.Users  
        .Project()  
        .To<UserViewModel>(parameters: new { }, membersToExpand: viewModel =>  
viewModel.Emails)  
        .FirstOrDefault();  
  
    if (user1 != null)  
    {  
        foreach (var email in user1.Emails)  
        {  
            Console.WriteLine(email.Text);  
        }  
    }  
}
```

این کوئری سبب خواهد شد تا صرفا خاصیت Emails از بانک اطلاعاتی واکشی شود و آدرس‌ها خیر. به این ترتیب می‌توان بر روی نحوه‌ی بارگذاری خواص راهبری کنترل کاملی داشت.

کدهای کامل این مطلب را [از اینجا](#) می‌توانید دریافت کنید.