## 附加实体到上下文

如果实体在数据库中存在，但是没有被上下文跟踪，可是使用DbSet.Attach方法将其附加到上下文，附加之后，实体处于Unchanged状态。处于Unchanged状态的实体不会参与SaveChanges的逻辑。

varexistingBlog = new Blog{ BlogId = 1, Name = "ADO.NET Blog"};

using(var context = newBloggingContext())

{

  context.Blogs.Attach(existingBlog);

  // Do some morework...

  context.SaveChanges();

}

设置DbEntityEntry对象的State属性，也可以附加对象到上下文中，如下：

varexistingBlog = new Blog{ BlogId = 1, Name = "ADO.NET Blog"};

using(var context = newBloggingContext())

{

  context.Entry(existingBlog).State =EntityState.Unchanged;

  // Do some morework...

  context.SaveChanges();

}

使用上述两种方法附加到上下文的实体如果还引用其他实体，那么这些实体也会被附加到上下文中，状态为Unchanged

使用如下方法附加一个存在于数据库，但是还没有附加到上下文的已修改实体：

varexistingBlog = new Blog{ BlogId = 1, Name = "ADO.NET Blog"};

using(var context = newBloggingContext())

{

  context.Entry(existingBlog).State =EntityState.Modified;

  // Do some morework...

  context.SaveChanges();

}

如果把一个实体的状态置为Modified，那么该实体的所有属性都将被标记为已更改状态，当SaveChanges被调用时，所有的属性值都将被保存到数据库。如果不想保存所有值，可以单独为每个想要修改的属性设置IsModified属性，如下：

using(var context = newBloggingContext())

{

    var blog =context.Blogs.Find(1);

    context.Entry(blog).Property(u =>u.Name).IsModified = true;

    // Use a stringfor the property name

    context.Entry(blog).Property("Name").IsModified = true;

}

如果该实体还引用其他未被跟踪实体，那么这些实体将会作为Unchanged状态的实体附加到上下文。如果想修改这些实体，只能单独把每个引用实体设置为修改状态。