先上文本，不知道为啥，wiki 发图多了不响应，图后期补上）

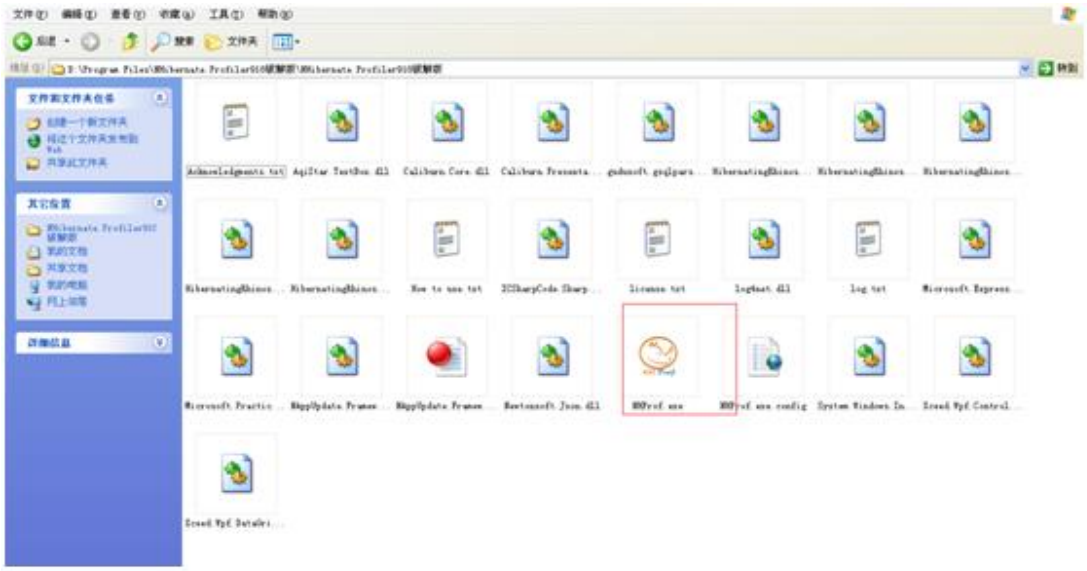
学习和使用 NH 有一段时间了，有一些皮毛上的研究和了解，打算写点如何来进行 NH 性能分析的分享,供新手和老鸟参考借鉴。

废话不多说，进 正题，我们知道 Nhibernate 是一款非侵入式的 Orm 框架，在业界也算是相当成熟了，相比 subsonic 而言，Nh 也还是相当重量级的，Nh 同时引入了一级和二级缓存的处理，当然 Nh 内部的优化策略做的是非常复杂的，我也无从得知，但是借助工具来查看 Nh 生成的 sql，仍然能感受 到 Nh 内部设计的优化和精妙所在。

今天主要来介绍下，如何在项目中来观察 Nhibernate 生成的 sql，怎么得知我在项目中写的代码性能究竟高不高呢,或者这些看上去没有问题的代码到底在实际执行的过程中，有没有问题。

其实很简单，就是工具的使用，全名叫 Nhibernate Profile，借助它，我们可以随时监测到 Nh 生成 sql 是什么样子的，当然了，看到了 sql，你自然也就知道你写的数据库操作代码到底性能如何了。
讲 Nh Profile 如何使用，很简单，如果是安装包，下载下来正常安装，官网地址 http://www.hibernatingrhinos.com/products/NHProf，在 wiki 上一篇 SaveOrUpdate()方 法异常的分享上已经给了 Nh 使用的一部分截图,地址见这里 http://192.168.10.90:81/CanYouWiki /index.php?doc-view-2622，那些黄色背景的图就是 Nh 使用的背景。官网下载的是需要注册码的。试用期一个月。我这里有 918 版本 的破解版，后面放出。

如果是破解版，使用更简单，把文件夹拷到一个磁盘，路径不限，随便放，要运行的 exe 如下图



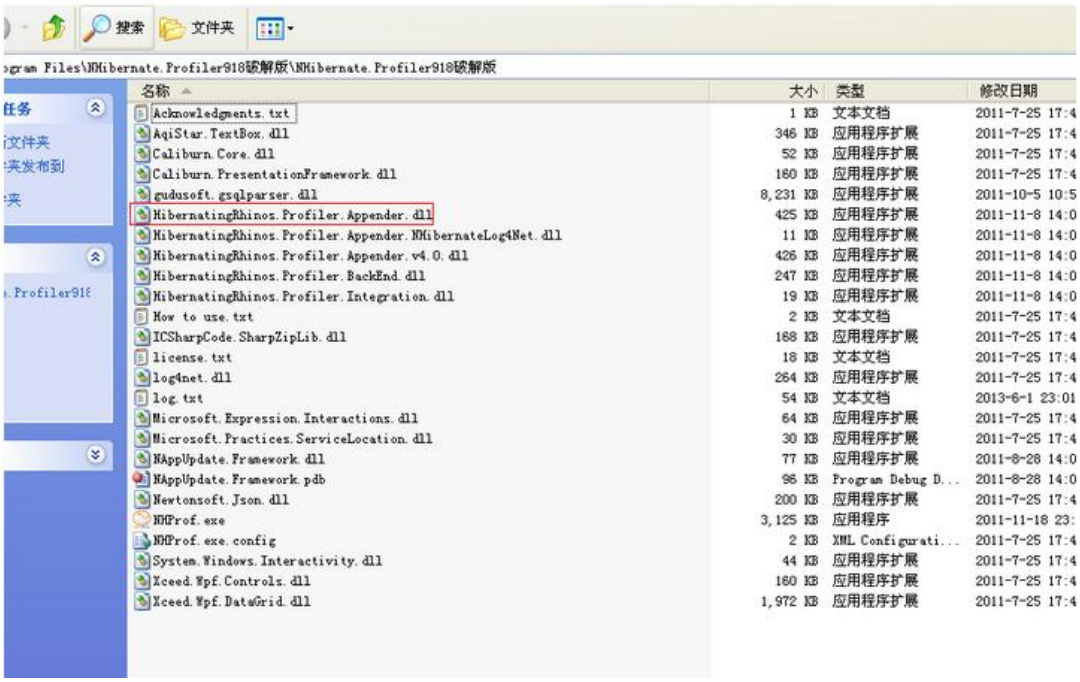
是不是说了很多废话，呃，好吧，程序的安装介绍已经结束，接下来介绍开发中如何配置使用，利用 Nh profile 来观察 nh 生成的 sql

我会拿 918 破解版的 Nh profile 作为示例来讲，应该不会有太大的区别，有问题欢迎私下找我讨论

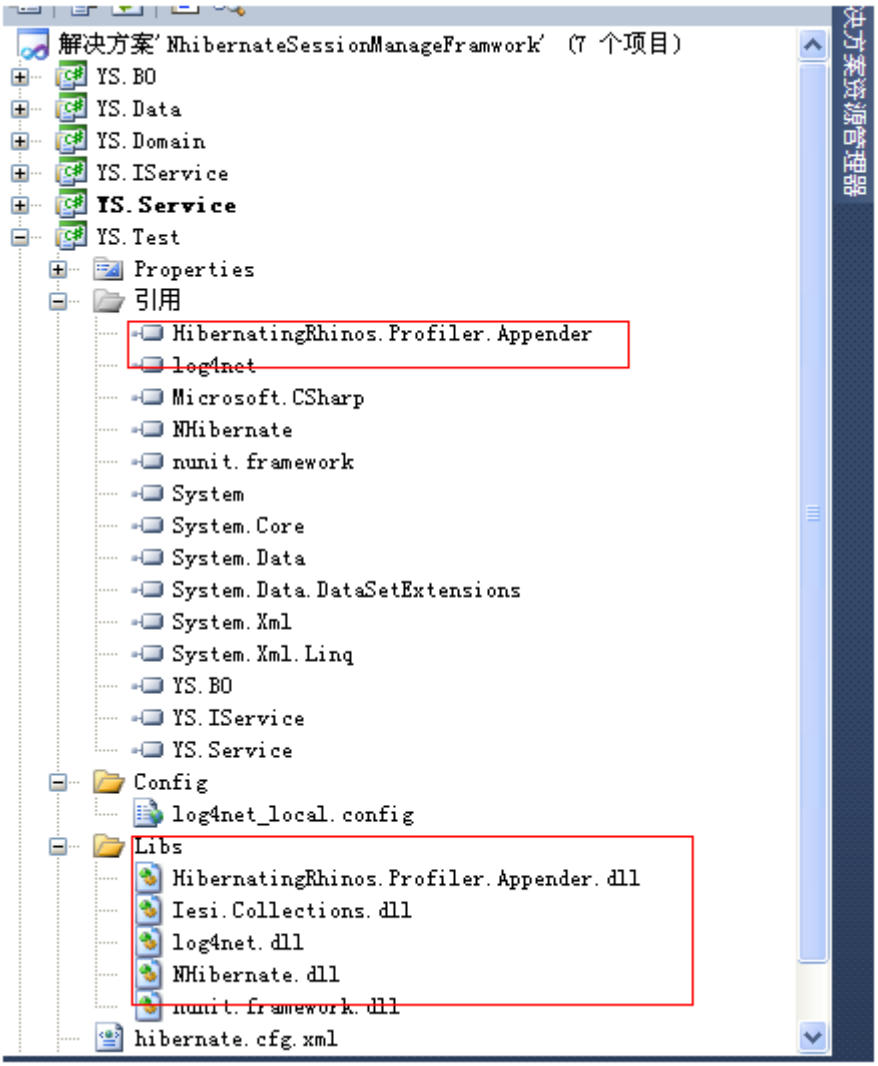
Nh profile 的工作，实际上就是利用 Log4Net 这个日志组件，将 Nh 生成的 sql 输出到 Nh Profile 的窗口上，当然这个是 Nh profile 最核心的功能，他还能看到很多数据库端的错误，能看到 Nh 内部在生成 sql 发生的错误等，还能观察到 Nh 内部 Session 的生 成，Session 的周期等等，能观察到所有向数据库发出的数据库操作，比如事务，存储过程调用，Insert，update，delete ，select 语句，我们观察到的这些语句都是经过 Nh 内部优化过的，以后再说这个优化问题。

听到这里，是不是感觉很心动，来动手做吧，亲眼见识一下就知道了。

配置 Nh Profile 我们需要引入两个组件，一个是 Log4net.dll,很熟悉吧，日志组件，公司用的很多，不介绍
了。另一个则是 Nh profile 提供的组件
HibernatingRhinos.Profiler.Appender.dll，如下图，在破解版文件包

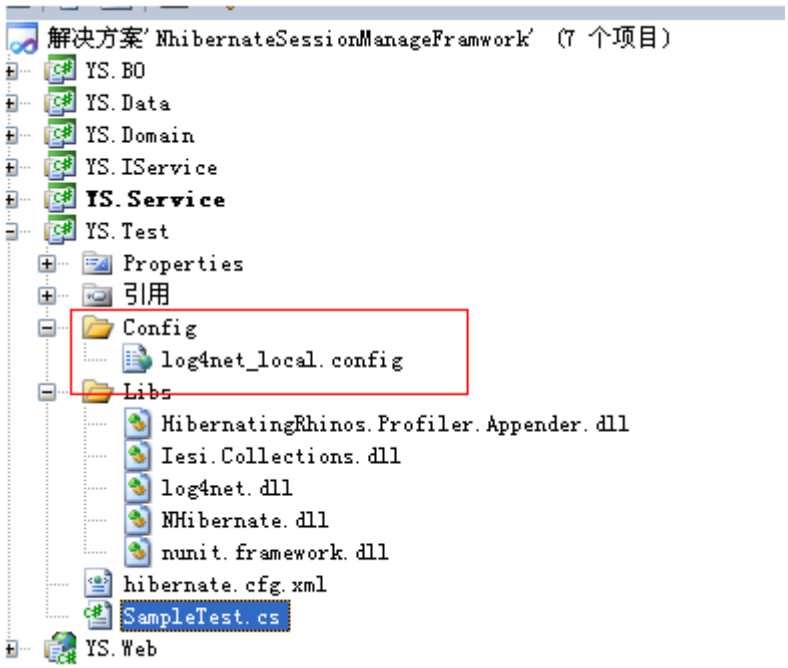


在项目文件中，将上面那两个 dll 考到一个 lib 文件夹中，引用他们，本处看到一个很简洁的框架，这与 Nh Profile 无关，哪怕只有一个层，一个再简单不过的 demo，放心，仍然可以使用它。

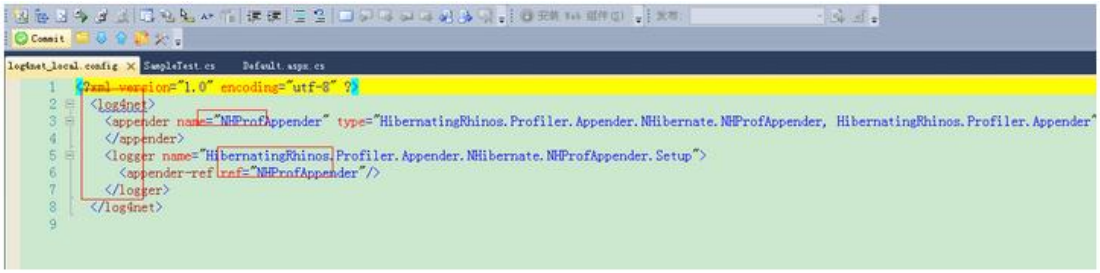


不 要把注意力转移到其他的几个 dll 上了 ， Iesi，Nhibernate，unit 这个三个 dll 是为我这个简单框架提供支持的，前面两个提供 Nh 的核心功能支持，Iesi 负责管理 Nh 中的集 合映射，包括包中的延迟加载等一些特性，Nhibernate.dll 不用说，提供 OO 的数据库操作到 sql 的生成和优化处理等工作，提供数据库访问的打 开和关闭等操作管理。说多了，以后有机会再介绍，回到 Nh profile 上来

说配置，也很简单，按规范来，建一个 config 文件夹，将放一个.config 文件进去，文件命名随便写，不要用中文



config 文件中只需要加如下配置代码



注意，根节点 log4net，appender 节和 logger 节不能写错，必须跟上面一摸一样，否则看不到效果的，我这里配置是配置在单元测试的那层的，所以会有新建 config 文件，配在 webconfig 中也是可以的，只不过我这简单起见就单独配了。不纠结。

好了，配置说完了，光这么配置还不够，我们还需要一个动作，就是启动的过程。跟开灯一样，布线了，通电了，灯要亮得拉开关。

nh profile 配置后需要进行初始化，才能完成达到我们能监测 Nh 的数据库操作的目的。

初始化代码如下，本处简单起见，将初始化代码写在了初始化事件里，用 TextFixtureSetUp 属性去修饰，在测试开始自动进行 Nh profile 的初始化工作

1,2,3 步，很清晰，大家注意我用红色圈住的东西就好了，别的代码不用关注，因为与 Nh profile 的使用没有关系
1 是获取 config 的路径

2 是拿路径来初始化 Log4net，这个初始化方式跟 Nh 的初始化方式几乎一模一样，可见软件在设计的时候，无论什么产品，设计思想没有变，相信大家从 Nh，linq to sql， PLinq-NH 的使用上也是能看到一些的。说多了...

3 步骤就是初始化 Nh Profile 了。没什么可讲的，本处 918 破解版按照上面没有注释的代码配置，完全可行。

上面讲了单元测试初始化环境的配置，那么我顺便简单讲以下完整的项目中的配置，首先在 web.config 中配置

定义和添加配置节，应该不用我讲了，不用去关注我注释掉的代码，那个是我 demo 中用过的，看我圈红的地方

初 始化的代码跟上面的一样的，写在 Global.asax 中的 Application_Start 事件中，同样要添加引用，当然如果你在 SessionFactory 的上一层再用 httpModule 来负责管理 Session 的话，你可以将初始化代码写在 Begin_Request 事件 里， 我个人建议写在这个事件中，后面我会给大家分享关于 OneSessionPerRequest 模式，这个模式的应用就要将 Nh profile 的初始化放在 Begin_Request 中，是最合适的。

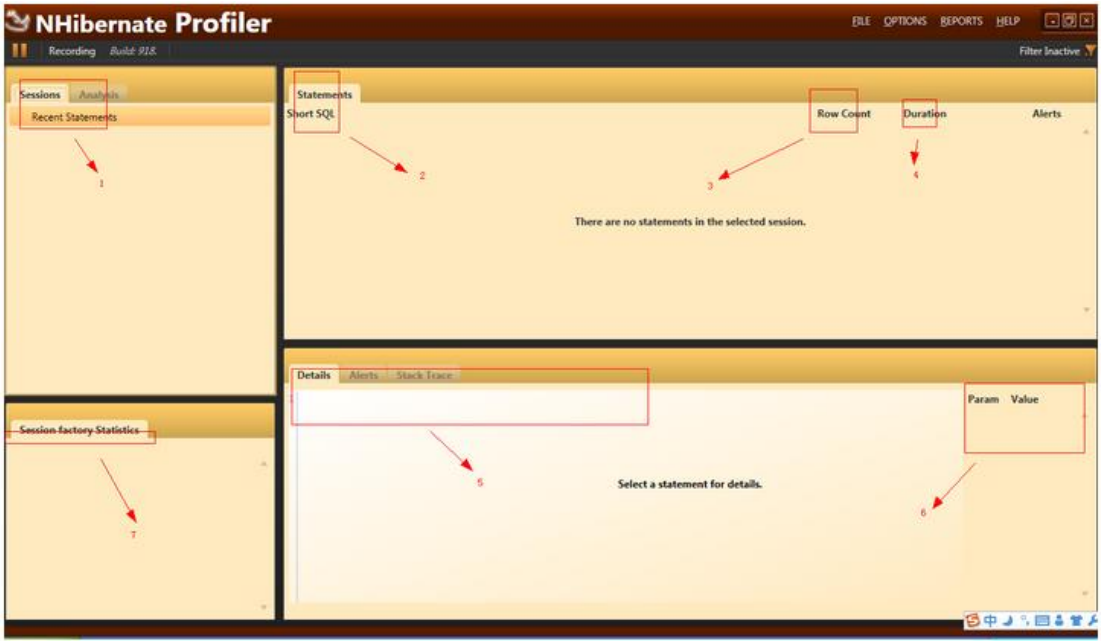
好了，配好了，初始化代码也写了，该拉灯了，注意，运行代码前一定要先将 Nh profile 跑起来，要灯亮必须先通电，囧.....。老样子，写个简单的单元测试，让程序跑起来，下图的 2，获取性别为男的数据集合，很简单吧，大家应该想象得到，生成的 sql 大家是个什么样子。接下来会上结果图

```
/// <summary>
/// 条件查询测试
/// </summary>
[Test]
public void GetAllPersonBySexTest()
{
    bool testResult = false;
    IList<PersonBo>personBos=personService.GetAllPersonBySex("男");

    if(personBos.Count>0)
    {
        testResult = true;
    }
    Assert.AreEqual(true,testResult);
}

//高性能批删除测试
[Test]
public void BatchDeleteByIds()
{
    bool testResult = false;
    IList<PersonBo> personBos = personService.GetAllPersonBySex("男");
    string[] ids=new string[personBos.Count];
    for (int i = 0; i < personBos.Count; i++)
```

2

打开在文章开头说的那个 Nh profile 的 exe 执行文件，下图红的那个名字，我发送快捷方式到桌面了..

NH Profile 运行起来初期是这个样子的



在上图中
1.Session 的显示窗口，NH 中每一个 Session 创建后，这里都会有显示，点击这个，右键，会关于 Session 记录的一些操作，比如删除，清除所有之类的

2.生成的 sql 语句，简短的，不用多说

3 生成的 sql 造成的数据库表受影响行数

4 处会有一个时间显示，应该是数据库操作时间，单位 ms。 这个地方会有两个数值，左边的代表数据库执行 sql 得到结果需要的时间，右边的是 nh 从发出 sql 到拿到结果集要的总时间。格外说一下，这个字不是固定不 变的，变化幅度很大。NH 内部和 db 内部有非常强大的以多线程和缓存为支撑的池管理功能，所以同样的方法，在首次运行和有效期内的第二次和第三次第 N 次显 示的时间不太一样，第一次的时间甚至会高出几百 ms，如果你观察到了这个，不用差异，不是代码的问题

5 处，这里是生成 sql 语句详情的显示窗口

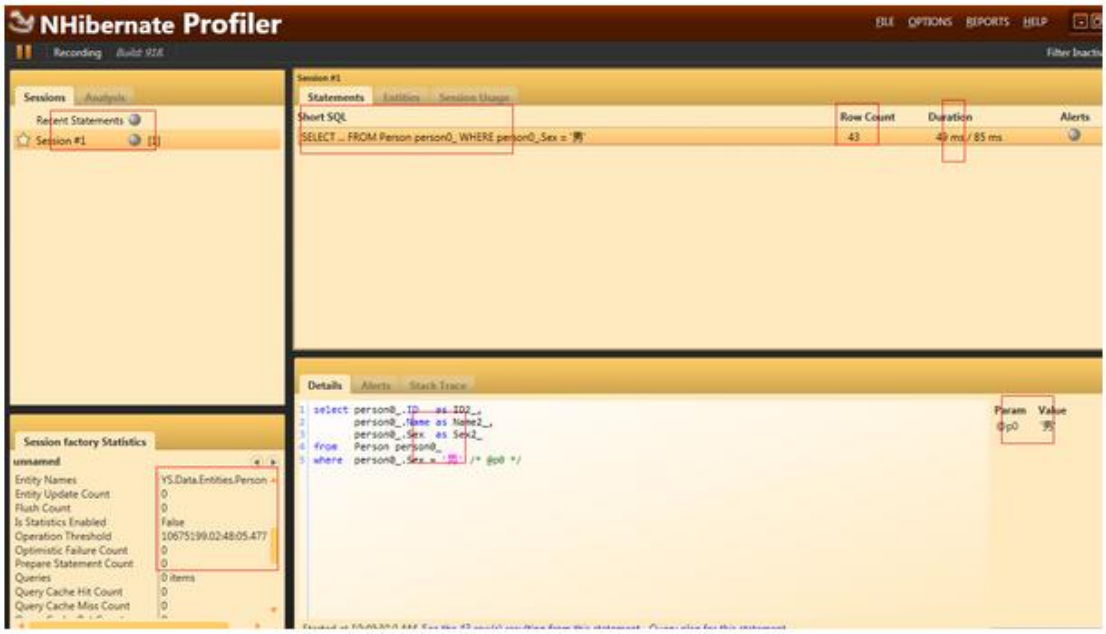
6 处，生成 sql 语句中的参数情况

7 处，这里显示的是 SessionFactory 中队本次 Session 管理的一些数据情况，一般我们不会去关注它，但是看看也是有用的，因为可以透过这里 查看到 Nh 内部 SessionFactory 对 Session 是一个什么样的管理情况，在一次数据库会话中，有哪些东西来负责管理了这次会话

补充：4 处右边有一个 alert，那个地方是一个状态标识，灰色代表正常，红色代表警告或者异常，可以查看详情

好了，关于 Nh profile 的窗口介绍完了

上图介绍实际我们观察到的数据，我运行我写的单元测试方法，Nh profile 显示如下



注意我标红色地方，成功运行了，nh 生成的 sql 是不是成功被我们捕获到了，有了这玩意，还担心我们看不到在用 Nh 的时候，不知道写的代码是否是高效和异常的么

凡是使用 Nh 作为 Orm 框架的，本工具可完美使用，不论你用的是 mysql ，ms sql Server，还是 oracle，对数据库操作性能有要求的你们，有什么理由拒绝使用这款工具呢。

当然，还有一款工具可是用来查看 nh 生成的 sql，如果你使用的 sql server，那么可是使用 sql server 自带的 profile，如下

这张图怎么都贴不上去，最后 wiki 的 mysql 服务器直接报出了 can not connect mysql 错误，说下怎么打开吧，

开始->ms Sql Server 2005->性能工具->Profile 登陆，你懂的。

这 个性能查看工具只对 ms sql server 数据库有效，而且我感觉不怎么好用，还是 Nh profile 好用些，它作为一种中间件，可以保证跨数据库的的数据监测，从 session 一产生就开始跟踪监测了，而且可以看到很多详情。注意，它跟 Fiddle

的工作是完全不一样的。

推荐项目经理在用 nH 作为 orm 框架的时候，使用这个工具，可以来约束开发人员来关注自己写的数据库操作代码的性能。

文章到这里就介绍完了，很晚了，希望对大家有帮助。大家在使用过程中碰到困难，可以私下来找我。
公司的 wiki 出了些问题，图一多提交就不响应了，这篇分享前后花了一些时间，前后为贴图 10 多次修改.....贴上来花的时间比写出来花的时间还多.....

以下放出 Nh profile 918 的破解版。不用去网上找了