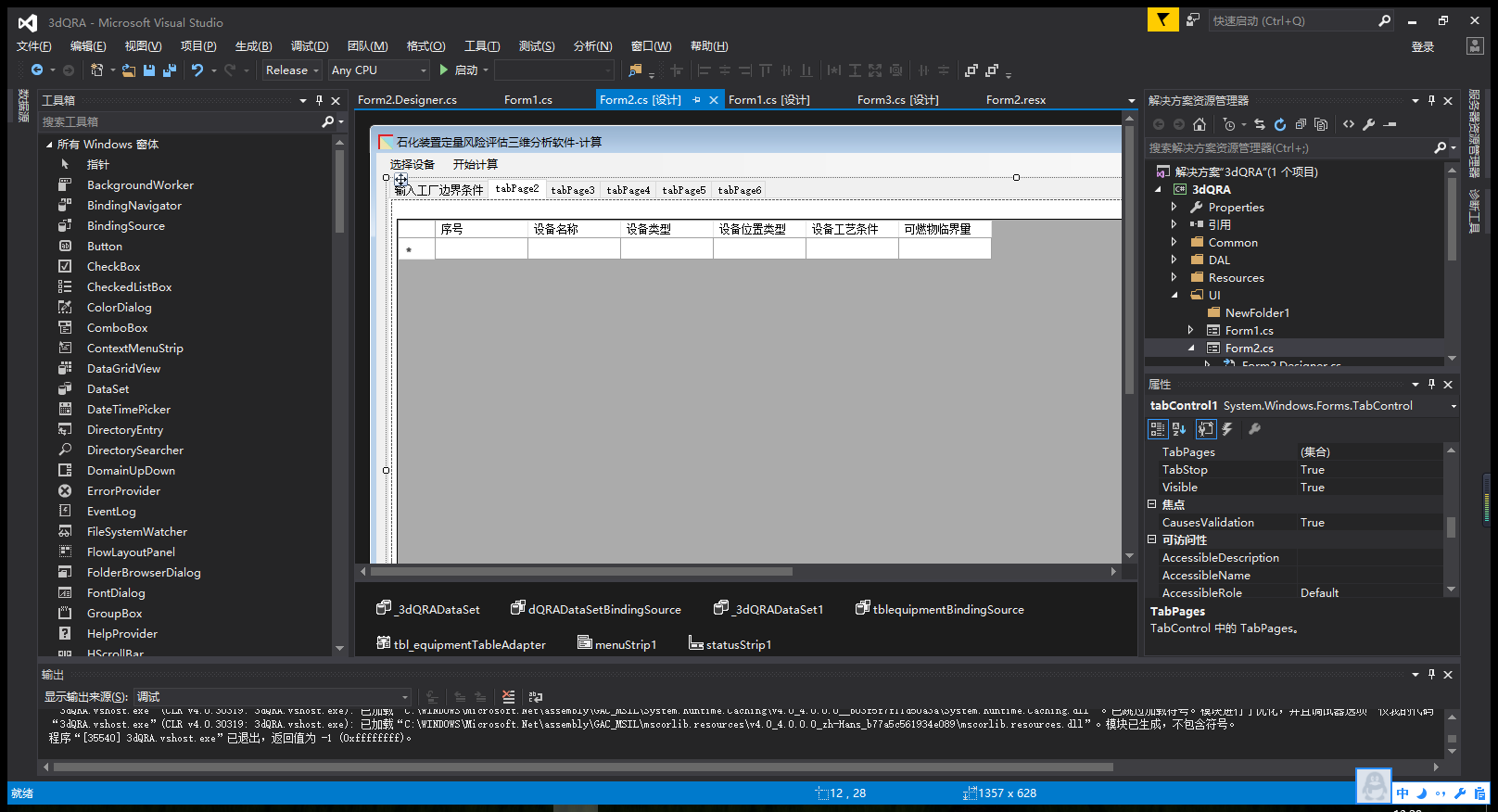


进入界面



编程画面

整个程序采用三层构架(3-tier architecture)就是将整个业务应用划分为：界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。分别放在UI、BLL和DAL三个文件夹下。

### 近期的工作：实现子功能——对于危险级别较高设备的选择，选择的指标为选择数。

整个子功能按照编程进行的顺序进行细分：

（1）输入边界

（2）显示边界（C#调用MATLAB）;

（3）边界检查（相交性检查）;

（4）输入设备（保存在数据库内）;

（5）设备检查（检查设备坐标是否在工厂内？）;

（6）对边界每隔50m进行划分;

（7）逐个点计算和设备之间的L（二维数组）;

（8）逐个设备计算指令数（一维数组）;

（9）计算选择数（由点为行和设备为列组成整个数表）;

（10）计算选择数的过程中，如果已经有与之前重复的设备，不记录入return项;

（11）显示计算选择数表和被选择出的设备（datagridview和数据库连接）。

整个功能进行到datagridview控件（就是程序中的表格）与数据库互联，由于最近都在搭建连接数据库的三层构架，所以在界面上工作量体现的不是很多，工作量体现在代码上。并且由VB6.0换成C#语言进行编写，代码的转换浪费了很多时间，这个转换的原因在于VB6.0对于数据库功能的支持较弱，并且程序界面符合扁平化潮流，更加美观。

### 为何要使用SQL SERVER2014 数据库？

1、统一保存结果。每个工厂或储油站的工作人员进行运行程序之后，工厂数学模型和风险评估计算结果保存于服务器的数据库中。

2、统一管理。由于风险评估结果统一保存于服务器中，从而服务器的管理员能够观察所有对各个工厂的情况，实现统一评价和统一管理。

3、数据提取。由于风险评估结果统一保存于服务器中，从而后续的程序使用人员可以调取其他工厂的数学模型，作为新的风险评估工作的参考。

（4、协同计算平台。多台电脑分别进行改变某一工厂或设备条件的多个算例，计算的结果保存在一起，方便计算结果的对比。进一步在该软件基础上深化开发，可以实现多计算机协同计算，用以完成基于风险考虑的工厂设计优化。）

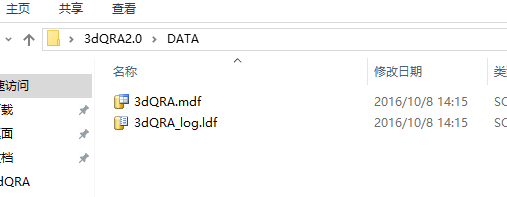
（5.进一步编程可以实现，省去安装软件的过程，加载网页，利用网页调用服务器的MATLAB完成计算过程，并将计算结果显示在网页上。这个过程一方面省去安装软件，另一方面软件使用平台将不受操作系统兼容性的限制。）。

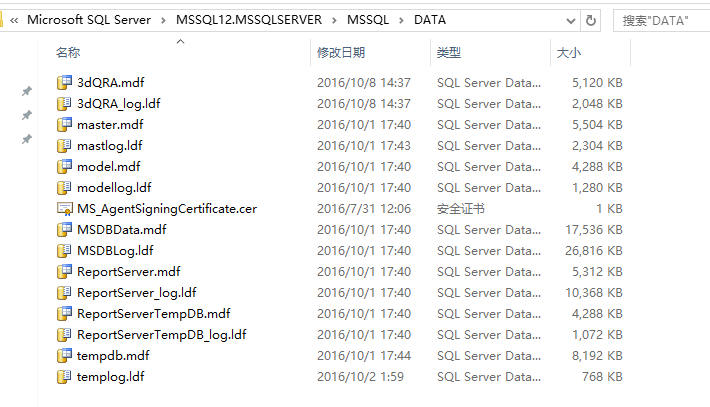
4.5两个，足够一个硕士从入学玩到毕业了。

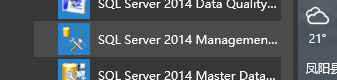
### 另外，使用前需要附加3dQRA数据库

流程：

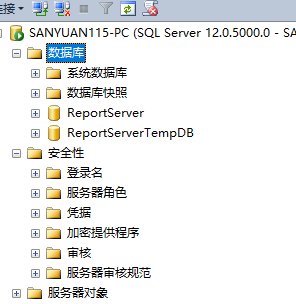
（1）先找到软件中下图中的两个文件



1. 拷贝到sql server2014的文件夹中，具体位置在：
2. 复制好之后，打开sql server2014，开始菜单里有很多个，你找到manage这一个；



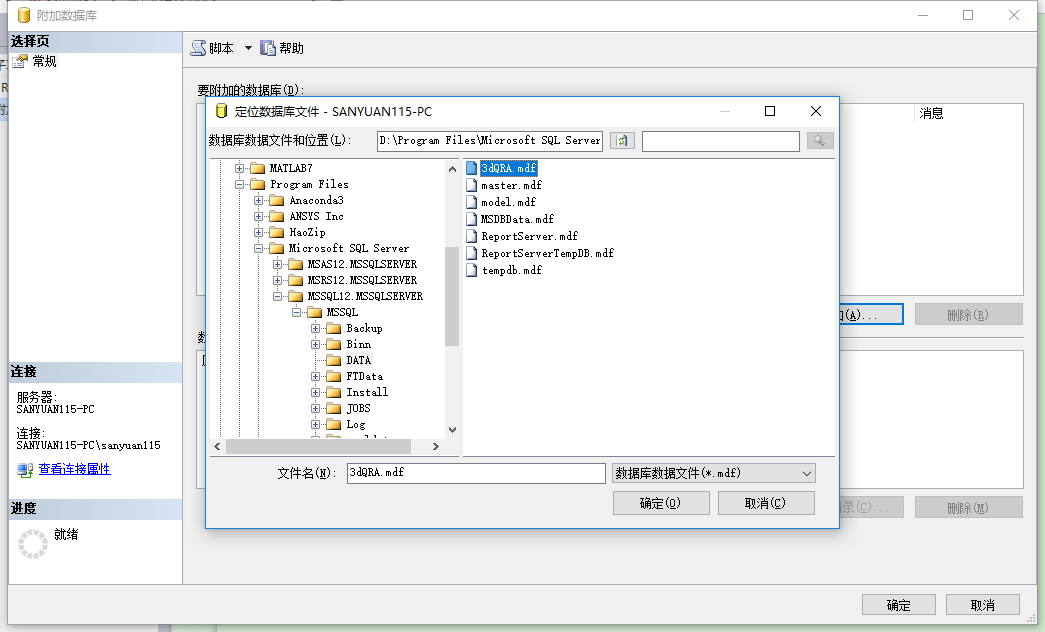
1. 右击数据库，点附加



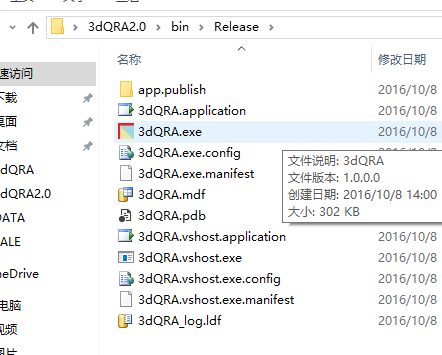
然后点添加



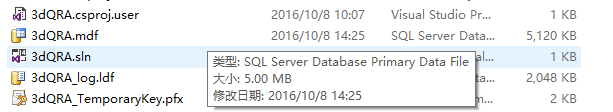
找到3dQRA,点确定即可



最后找到程序中的3dQRA.EXE运行程序。



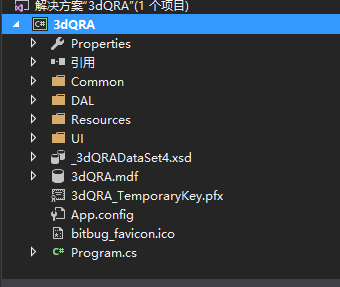
源代码可以通过程序文件夹中的3dQRA.sln文件查看



最近，我的软件方面的工作集中在程序与数据库的连接，因此功能实现在界面层面表现不多，主要在数据访问层，最终实现的功能可以将程序与数据库链接在一起。mdf文件为数据库文件，里面包含着大量相关的表格，这些表格中储存着计算使用到的石化设备信息、储油站的边界、建筑和人员信息，以及计算结果，即人员风险在储运站空间中的分布。由于这些信息较为庞杂，编程时考虑的问题也就需要很全面。

使用数据库可以将大量的信息有序的排列在一起，我们以前计算时总是忘记总结，导致同一个算例重复计算浪费时间，使用数据库就是有这个优势，计算后的结果有条理的放在一起，想查找以前计算的结果也很方便。并且大量的数据如果用其它的存储方式存取比较慢，这也是用数据库的一个优势。假如有师弟师妹有兴趣，也可以在我的程序基础上进行修改实现流体的并行计算，也是一个新的科研方向。

为了实现数据库与程序计算的链接，整个程序采用三层架构编写，即界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。业务逻辑层放置了一些需要重复调用的函数，数据访问层放置了一些数据库与程序链接的函数，界面层就是用户见到的软件界面。虽然在界面中看不到，但这些工作是编程中的轮子，造轮子是造车比不可少的过程。这三个层分别放在



最近，我的软件方面的工作集中在程序与数据库的连接，因此功能实现在界面层面表现不多，主要在数据访问层，最终实现的功能可以将程序与数据库链接在一起。mdf文件为数据库文件，里面包含着大量相关的表格，这些表格中储存着计算使用到的石化设备信息、储油站的边界、建筑和人员信息，以及计算结果，即人员风险在储运站空间中的分布。由于这些信息较为庞杂，编程时考虑的问题也就需要很全面。

使用数据库可以将大量的信息有序的排列在一起，我们以前计算时总是忘记总结，导致同一个算例重复计算浪费时间，使用数据库就是有这个优势，计算后的结果有条理的放在一起，想查找以前计算的结果也很方便。并且大量的数据如果用其它的存储方式存取比较慢，这也是用数据库的一个优势。假如有师弟师妹有兴趣，也可以在我的程序基础上进行修改实现流体的并行计算，也是一个新的科研方向。

为了实现数据库与程序计算的链接，整个程序采用三层架构编写，即界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。业务逻辑层放置了一些需要重复调用的函数，数据访问层放置了一些数据库与程序链接的函数，界面层就是用户见到的软件界面。虽然在界面中看不到，但这些工作是编程中的轮子，造轮子是造车比不可少的过程。这三个层分别放在