# 功能设计&UI

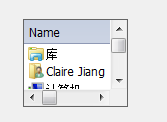
基础功能

1. 可以选择要检测的设备类别
2. 保存检测结果
3. 发送检测结果
4. 显示检测结果

预期功能

1. 选择保存检测结果的位置

mfcshelllist control



1. 除设备检测外，还可输入其他备注，完成所有测试结果填写动作在APP端完成，不需要登录网站
2. PCI device 给出拓扑结构
3. Disk部分，按照盘符显示，每个盘符对应型号厂商

# 日志

1.18

* 完成：

1. 完成PCI配置空间读取和输出到文件

* 明日：

1.19

* 完成：

1. PCI读配置空间的bug
2. 读memory，graphics card，disk，network的代码

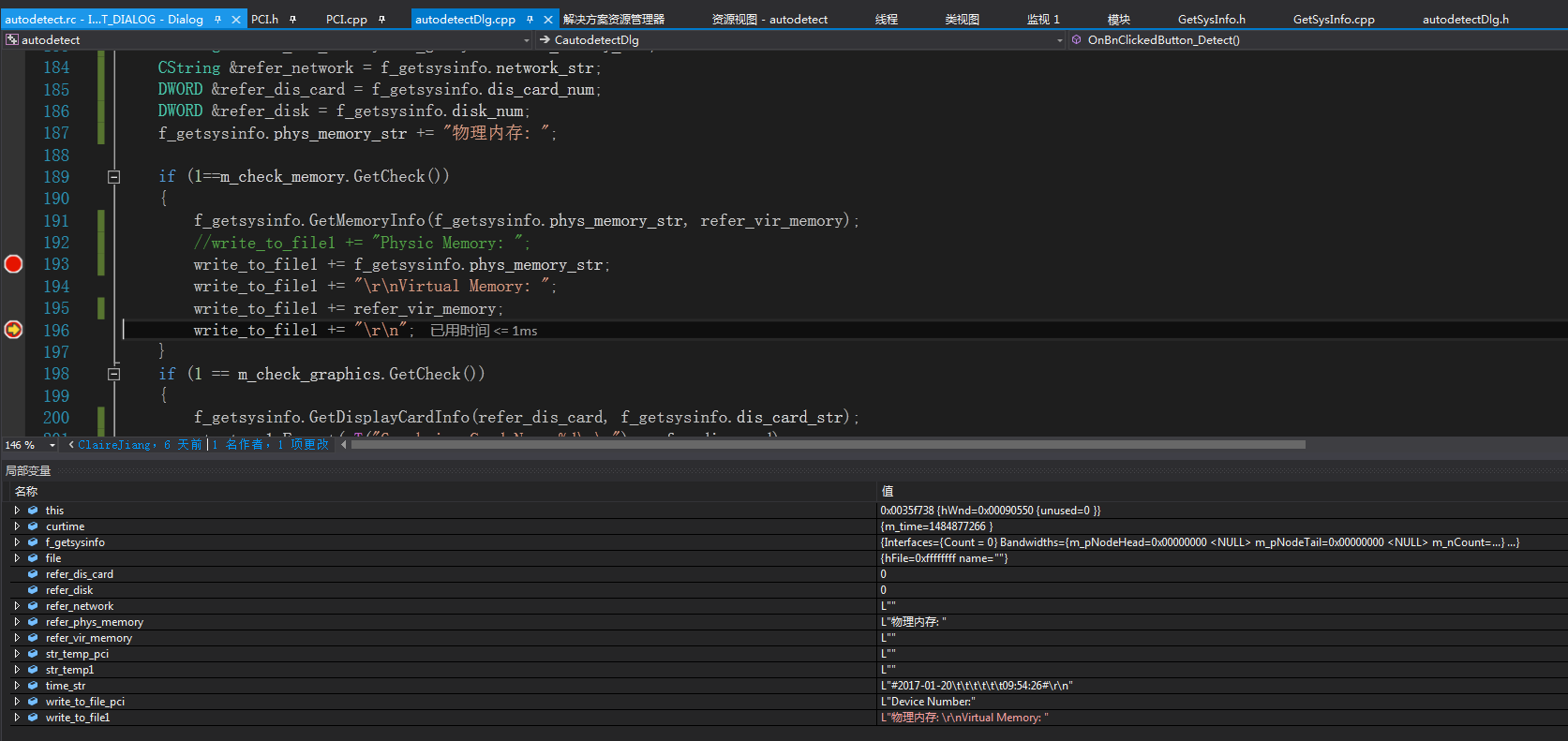
* 明日：

1. 解决读memory等部分代码的bug

1．20

* 完成：

1. 读memory等部分代码的bug



1. 在getmemoryinfo里保存了string，但是出了这个函数就没了。

原因：有中文字符，改成英文就没事儿了

1. 没有进入disk，网卡的函数，显卡的名字string有一部分乱码

原因：不是没有进入函数，中间return了

* 明天：

1. 解决其他有bug的部分

1.22

完成：

1. 显卡乱码

Step1：从HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\Enum\PCI\VEN\_8086&DEV\_1912&SUBSYS\_30D217AA&REV\_06/读到的有乱码的部分，HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\services\igfx\Device0这个表里的没有乱码

HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet002\services\igfx这还有一个igfx

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\services\igfx这还有一个igfx

如果是AMD显卡，位置在HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\services\amdkmdap

Gfx独立显卡，HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Intel\GFX

Gpp交火

Igfx集成显卡

Pci pci显卡

1. Disk厂家型号

Step1：的信息也读出来了，但是没有保存到函数参数里，另外比较disk个数的代码是干什么用的搞不明白，这段code导致了提前return——》disk部分读到的信息不是我想要的，没有硬盘型号，厂家，只有盘符。disk的型号厂家在注册表system/currentcontrolset/services/disk/enum——》厂家 型号

1. 网卡扫出9个设备，只有2个是我要的

Step1：网卡在注册表hkey local machine/software/microsoft/windows NT/currentversion/networkcards——》

1. 在CHX板子问题：内存16G只显示4G（用CPU-Z可以读到正确的内存容量，但是读不到厂家型号）；显卡显示amdkmdag
2. 有用的注册表项：
3. HKEY\_LOCAL\_MACHINE\HARDWARE\DESCRIPTION\System\BIOS

* 明天：

1.23

* 完成：

1. BIOS信息读取
2. Physic Disk的厂家型号
3. 添加完善UI

* 明天

1. 补上OS部分
2. 除了send按键，完成功能实现

1.24

* 完成：
* 明天：

2.6

* 完成：

1. UI里除了send都完成了

* 明天：

弄好使

2.7

* 完成：

1. 新UI好使了
2. UI新增PCI地址填写

* 明天：

1. PCI地址部分，写注册表

备注：

1. 子对话框添加初始化函数，afx clicked send函数什么鬼

2.8

* 完成：

写注册表的一部分，不完善

* 明天

完善PCI地址的部分（读写注册表）

备注：

手残＋脑残 没救了

2.9

* 完成

PCI地址部分（读写注册表）

USB搞出一部分

* 明天

USB完善

2.10

* 完成：

USB能读出历史记录

独显的namestring读取

* 明天：

2.13

* 完成：

用github管理代码

* 明天：

用github管理代码

2.14

* 完成：

用github管理代码

* 明天：

Rest

|  |  |
| --- | --- |
| PART | Question |
| MEMORY | 1. 大小不对 2. 型号没有 3. 插槽无法得出 |
| DISK(SATA) | 1. 得到的字符串太长，有些不必要的字符 2. 还是会导致hang 3. 插槽不能确定，可能和PCI重复？ |
| Graphics Card | 1. 我们的主板集显不在注册表里留信息 |
| BIOS |  |
| OS | 1. 暂时空 |
| USB | 1. 读出来的是历史记录 |
| PCI | 1. 地址写到注册表里，以后都默认 |
| Network | 1. 得出很多不需要的 |
| UI | 1. Send发送 2. 丑，不整齐 |

# Topic

注册表

Currentcontrolset：运行时配置

Controlset001：真实配置

Controlset002：运行时配置

[HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\Services\TermService]  下记录着“系统服务”的相关设置。      
那么ControlSet001、ControlSet002、CurrentControlSet有什么区别呢？  
          默认情况下：  
          ControlSet001：系统真实的配置信息。  
          CurrentControlSet：运行时配置。  
windows启动时会从ControlSet001复制一份副本，作为操作系统当前的配置信息。  
我们对于计算机配置所作的修改都是直接写入到 CurrentControlSet，在重启过程中，windows会用CurrentControlSet  
的内容覆盖掉ControlSet001，以 保证这两个控件组一致。  
          ControlSet002：最近一次成功启动的配置信息。  
 当操作系统每成功启动一次（指成功登录），它都将CurrentControlSet和ControlSet001中的数据复制到   
ControlSet002中。  
----------------------------------------------------------------------------  
          但是，这个顺序和数目不是一成不变的，改变就发生在使用过“最近一次的正确配置”之后。这个时候，系  
统会把002当作系统真实的配置信息，  
而001这个存在问题的控件组会被备份封存起来。  
          [HKEY\_LOCAL\_MACHINE\SYSTEM\system\select]  下记录着下次将要使用的配置:  
　　“Current”数据项目表示 Windows在这次启动过程中使用的控件组。    
　　“Default”数据项目表示 Windows 在下次启动时将使用的控件组，它与这次启动使用的控件组相同。    
　　“LastKnownGood ”数据项目表示您在启动过程中选择“最近一次的正确配置”时 Windows XP 将使用的控件组。    
　　“Failed”数据项表示 Windows XP 在其中保存失败启动产生的数据的控件组。 此控件组在用户第一次调用“最近  
一次的正确配置”选项之前并不实际存在。