

Non-psychic debugging: If somebody's complaining that a collection should be empty but isn't, you might want to see what's in there

 devblogs.microsoft.com/oldnewthing/20140131-00

January 31, 2014



Raymond Chen

A programmer on the GHI team (yes, the same GHI team) reported that they were hitting an assertion failure using an internal library and asked for help debugging it.

```
// All factories should be unregistered by now
assert(m_pFactoryList->IsEmpty());
```

“Can somebody help me figure out which factory it is that did not get unregistered?”

I didn't work on this internal library, but on the other hand I'm not afraid to look inside.

Let's see what a `m_pFactoryList` looks like.

```
0:000> ?? this->m_pFactoryList
class LookupTable<CFactory*>
  +0x000 m_uListSize      : 1 // this probably means that the list has one element
  +0x004 m_pList         : 0x00212e60 LookupTable<CFactory*>::ENTRY // this is
probably the list
  +0x008 m_uCapacity     : 0x7f
0:000> ?? this->m_pFactoryList->m_pList
struct LookupTable<CFactory*>::ENTRY * 0x00212e60
  +0x000 pszName        : 0x02cf4048 "GHI_factory"
  +0x004 data           : 0x02cf4ce0 CFactory* // I bet this is the item that got
leaked
0:000> dps 0x02cf4ce0 l1
02cf4ce0 6b626d58 GHI!CGhiFactory::`vftable`
```

No psychic powers needed here. I just followed my nose.

The assertion says that a list is not empty. Therefore, we should look to see what is on the list.

As a general rule, code is not intentionally written to be impossible to understand. The person who wrote it meant well, so if you see a member called `m_uListSize`, it's a pretty safe bet that it represents the list size. And if you see a member called `m_pList`, it probably points to the start of the list.

Raymond Chen

Follow

