THE TORQUE MEMORY MANAGER

1 Setting Up

Finished Code:

```
function startexercise004()
{
   $prepFile = expandFilename( "~/baseLineTMMDump" );
   $TMMFile = expandFilename( "~/debugTMMDump" );
}

function doPrepWork()
{
   flagCurrentAllocs();
   dumpUnflaggedAllocs($prepFile);
}

function doTMMDump()
{
   dumpUnflaggedAllocs($TMMFile);
}
```


2 Evaluating Dumps

Output From tmmTest1():

(Note: Your output may vary slightly.)

```
==>tmmTest1();
****** PRE WORK ************


****** POST WORK *********
h:\00_currentwip\bk002_gpgt2\tge_1_5_2\engine\console\consoleinternal.cc^297^16^64749
h:\00_currentwip\bk002_gpgt2\tge_1_5_2\engine\console\consoleinternal.cc^301^60^64750
h:\00_currentwip\bk002_gpgt2\tge_1_5_2\engine\console\consoleinternal.cc^483^24^64748
h:\00_currentwip\bk002_gpgt2\tge_1_5_2\engine\console\consoleobject.h^366^56^64747
```

Questions:

1. Yes, it does. Any allocations except "consoleinternal.cc" should be considered suspect.
2. The problem above has the following attributes:
   - File: "engine\console\consoleobject.h"
   - Line: 366
   - Allocation Size: 56 bytes.

The specific code that is 'leaking' is:

```
ConsoleObject* create() const { return new T; }
```

As can be seen, this is in fact an allocation, just as we would expect. Of course, in this case, the problem is in our scripts (and logic) and not in C++. So, my advice is to consider all options and not to assume immediately that all unflagged allocations are C++ leaks.

How do I know that the problem is in our scripts? Recall that tmmTest1() does not delete the SimObject, in order to simulate a leak. This means the allocated memory will show up in our dump.

