Subject: sMutexLock implementation
Posted by hojtsy on Thu, 22 May 2008 08:38:33 GMT
View Forum Message <> Reply to Message

In Mt.cpp there is

```
Mutex& sMutexLock()
{
 static Mutex *section;
 if(!section) {
  static byte b[sizeof(Mutex)];
  section = new(b) Mutex;
 }
 return *section;
}
```

How is this different from the simpler

```
Mutex& sMutexLock()
{
 static Mutex m;
 return m;
}
```

In both cases the Mutex constructor will be called when the function is first called. In both cases the function needs external protection from MT race conditions.

On a side note, this function is not on the interface (Mt.h), why not make it file static in Mt.cpp to avoid name clashes?