

---

Subject: Core multithread dangers

Posted by [hojtsy](#) on Fri, 03 Feb 2006 22:09:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I was told before that Core & Web packages are thread-safe. But some methods of Core are not thread-safe, and something like

CriticalSection::LockMain should be added to them to fix this. Here is a list of them.

static TimingInspector& s\_zero()

TimingInspector::TimingInspector(const char \*\_name)

static void sInit()

String& sHomeDir()

String ConfigFile(const char \*file)

byte \*Alloc4KBRaw()

void \*MemoryAllocPermanent(size\_t size)

void \*MemoryAlloc(size\_t size)

void MemoryProbe(const char \*name, dword flags)

const LanguageInfo& GetLanguageInfo(int lcode)

static Array<LngModule>& sMod()

CriticalSection& MainCriticalSection()

PageInfo \*NewPage(int magnitude)

static CriticalSection& sHeapLock()

static CriticalSection& sHeapLock2()

static inline void sInitTables()

void MemoryInitDiagnostics()

RHITCOUNT(n)

static int sMappingGranularity\_()

VectorMap<String, VectorMap<String, VectorMap<String, Topic > > >& TopicBase()

String GetIniKey(const char \*name)

These other functions of Core need bigger rewrite to be threadsafe:sDumpPtr(void \*ptr)

sDump(dword w)

const char \*Dump(PageInfo \*page)

const char \*MemoryCounters()

These functions of the Web package are not thread-safe:

void HttpServer::ReadPostData(Socket& socket, HttpQuery& query)

void SSLInit()

static const dword \*GetCRCTable()

RefPtr<HttpQuery::Data> HttpQuery::Empty()

dword WINAPI HttpExtensionProc(EXTENSION\_CONTROL\_BLOCK \*ecb)

The problem is always with the local static variables which are not protected from parallel initialization on two threads.

---