Subject: DnD hangs in MT Refresh()ing
Posted by kohait00 on Thu, 28 Jan 2010 15:58:35 GMT
View Forum Message <> Reply to Message

hi people

i am currently struggeling with a GUI / MT behaviour of U++ Ctrl stuff. i attached a Test environment, a modified CoWork. it does the following.

it fires up some CoWork Threads/Jobs, which simply do a

```
while(dorun)
{
  Refresh(); //which internally does GuiLock somewhere, no problem, THEORETICALLY
  Sleep(1);
}
```

and the Paint paints something dynamic, to see if it still is running something.

there is also a TreeCtrl, filled in same way as TreCtrlDnd example,

PROBLEM/BUG: start DnD some elemen of TreeCtrl without releasing it, soon the running Threads will freeze and not perform any refresh anymore. if minimizing and maximizing again, it starts to work, having performed its work; >> but until than it hangs !!!

do i misunderstand somehow the concept of GuiLock and MT things here? it seems to me that its kind of a deadlock with maybe one of the sPteLock, StaticMutex or GlobalMutex or the like..any guess?

BACKGROUND:

i am working on a application that uses I/O Completion Ports Queue (Win32 specific) and a CoWork threads pool, which performs he completion tasks. i receive periodic, frequent data, which i directly push throough to controls, which should be ok, since they somewhere invoke Refresh() which will do a GuiLock. so tey should be ok. and it works, that far. as soon as i start to DnD things, while a control, which is beeing driven directly by the completion threads for refreshment, is open, the whole thing hangs/freezes without mercy. so i tried to make a simple test. which almost does the same, but my own application is not recoverable. if i stop the debuger it points me to some point in ntdll, which smells of Mutex or Critical Section or what.

File Attachments

1) CoWork.rar, downloaded 567 times

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Thu, 28 Jan 2010 22:55:46 GMT
View Forum Message <> Reply to Message

How many cores do you have?

I have tried the example; D&D just slows down the app. Maybe the problem is related to the number of cores? (4 here..)

Needs more investigation.... Right now, I have no clue, party because I cannot replicate. But maybe I will try to find out whether the slowdown is OK - perhaps I will get a clue on the way.

Mirek

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Thu, 28 Jan 2010 23:22:51 GMT
View Forum Message <> Reply to Message

2 cores here, one needs to playe something with the element, dragging it in the tree without release, it does not hang immediately. slows down, later it hangs. sometimes it hangs as soon as released.

this is really weired. i was trying to investigate what kind of mutexes are involved. there is the NonMainMutex, sGLock, and also one needs to consider the semaphore wait in ICall interface. maybe the threads just inject themselves at a point when calling Refresh and locking GuiLock, not beeing able to continue doe to wait for semaphore of ICall, which is released after completion of ICall in main thread, which in turn cant complete the call, cause the DragAndDrop thing is doing a release LeaveGMutexAll, performing the draganddrop and not beeing able to EnterGMutex(level) anymore, cause one of the worker threads has started something. but there you also do a leave/wait for semaphore/enter, so GuiLock is not aquired there. man, no idea

i'll watch on that, thanks

BTW: have you spent any thoughts on integrating the high performance stuff of completion ports for sockets/files, both on win32 and linux (real kernel aio, not the posix indermediate rubish)?

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Fri, 29 Jan 2010 07:45:36 GMT
View Forum Message <> Reply to Message

to reproduce it, hopefully 100 percent:

start dragging the tree element, drag a bit, dont release and dont move it anymore, still holding. after a time the whole stuff freezes, well after minimizing its all ok, though, but in my software it is just blocked

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Fri, 29 Jan 2010 08:32:59 GMT
View Forum Message <> Reply to Message

got some investigation on it (on MY app, not the CoWorker, but should be similar, i checked it, it behaves like that):

the problem even occurs with only 1 worker thread
(replacing the whole CoWork co with a Thread th and Run() the DoRefresh cb there)

DoCall() in Win32Wnd.cpp:747 always returns true (because NonMain counter is up to 25), so the ProcessEvent(bool *quit) cant go on processing other stuff and do a repaint on gui or dispatch user events.

the sGLock is owned by GUI thread, GuiSleep is running and so releasing GUI mutex sGLock from time to time, so potential other threads in need of gui lock, should run right.

the NonMainLock is owned by a worker thread and is not released anymore, there comes the deadlock, but NOT with the gui mutex, for sure, maybe the semaphore.wait in ICall or Call interface.

What is the NonMainMutex actually for? i think that is more or less the problem..

any idea?

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Fri, 29 Jan 2010 09:31:09 GMT
View Forum Message <> Reply to Message

one more:

indeed, in CtrlMT.cpp
void Ctrl::ICall(Callback cb)

is suffering a not release of semaphore due to an early CtrlCall.Clear in a DoCall somewhere in between, maybe from another thread??, so that the PerformCall cant be executed, which would release the semaphore...is there any other thread accessing DoCall?

a short log, sem1 is the ones from ICall, simpli with an id counter, sem2 is the ones from Call interfaces.

..

DoCall
Sem release, sem1 934 sem2 0
DoCall::CtrlCall clear sem1 934 sem2 0
SEM1 released 934

```
SEM1 wait 935
DoCall
Sem release, sem1 935 sem2 0
DoCall::CtrlCall clear sem1 935 sem2 0
SEM1 released 935
SEM1 wait 936
DoCall
Sem release, sem1 936 sem2 0
SEM1 released 936
DoCall::CtrlCall clear sem1 936 sem2 0
SEM1 wait 937
DoCall
DoCall::CtrlCall clear sem1 937 sem2 0
DoCall
DoCall::CtrlCall clear sem1 937 sem2 0
DoCall
DoCall::CtrlCall clear sem1 937 sem2 0
```

the CtrlMt.cpp used for that is the current one, from yesterday svn, so you can see where what is logged

```
#include "CtrlCore.h"

#define LLOG(x) // LOG(x)

NAMESPACE_UPP

#ifdef _MULTITHREADED
static int sem1 = 0;
static int sem2 = 0;

static int         NonMain;
static StaticMutex NonMainLock;

void EnterGuiMutex()
{
 Mutex & m = NonMainLock.Get();
 LLOG("Thread " << IsMainThread() << " trying to lock");
 bool nonmain = !IsMainThread();
 if(nonmain)
  NonMainLock.Enter();
 EnterGMutex();
 if(nonmain)
  NonMain++;
 LLOG("Thread " << IsMainThread() << " LOCK");
}
```

```
void LeaveGuiMutex()
{
 Mutex & m = NonMainLock.Get();
 LLOG("Thread " << IsMainThread() << " trying to unlock");
 bool nonmain = !IsMainThread();
 if(nonmain)
  NonMain--;
 LeaveGMutex();
 if(nonmain)
  NonMainLock.Leave();
 LLOG("Thread " << IsMainThread() << " UNLOCK");
}

struct Ctrl::CallBox {
 Semaphore sem;
 Callback  cb;
};

void Ctrl::PerformCall(Ctrl::CallBox *cbox)
{
 cbox->cb();
 LOG("Sem release, sem1 " << sem1 << " sem2 " << sem2);
 cbox->sem.Release();
}

Callback    Ctrl::CtrlCall;

void WakeUpGuiThread();

bool Ctrl::DoCall()
{
 LOG("DoCall");
 CtrlCall();
 LOG("DoCall::CtrlCall clear sem1 " << sem1 << " sem2 " << sem2);
 CtrlCall.Clear();
 LLOG("--- DoCall, nonmain: " << NonMain);
 int a = NonMain;
 return NonMain;
}

void Ctrl::ICall(Callback cb)
{
 LLOG("Ctrl::Call " << IsMainThread() << ", nonmain: " << NonMain);
 if(IsMainThread())
  cb();
 else {
  CallBox cbox;
  cbox.cb = cb; ++sem1;
```

```
  CtrlCall = callback1(PerformCall, &cbox);
  int level = LeaveGMutexAll();
  WakeUpGuiThread();
  LLOG("Waiting for semaphore");
  if(!Thread::IsShutdownThreads())
  {
   LOG("SEM1 wait " << sem1);
   cbox.sem.Wait();
   LOG("SEM1 released " << sem1);
  }
  EnterGMutex(level);
 }
 LLOG("-- Ctrl::Call " << IsMainThread());
}

void Ctrl::Call(Callback cb)
{
 if(IsMainThread())
  cb();
 else {
  CallBox cbox;
  cbox.cb = cb; ++sem2;
  UPP::PostCallback(callback1(PerformCall, &cbox));
  int n = NonMain;
  int nn = n;
  NonMain = 0;
  for(int i = 0; i < n; i++)
   NonMainLock.Leave();
  int level = LeaveGMutexAll();
  WakeUpGuiThread();
  if(!Thread::IsShutdownThreads())
  {
   LOG("SEM2 wait " << sem2);
   cbox.sem.Wait();
   LOG("SEM2 released " << sem2);
  }
  for(int i = 0; i < n; i++)
   NonMainLock.Enter();
  EnterGMutex(level);
  NonMain = n;
 }
}

#else

bool Ctrl::DoCall()
{
 return false;
```

```
}

void Ctrl::ICall(Callback cb)
{
 cb();
}

void Ctrl::Call(Callback cb)
{
 cb();
}
#endif

void Ctrl::GuiSleep(int ms)
{
 Call(callback1(&Ctrl::GuiSleep0, ms));
}

void Ctrl::WndDestroy()
{
 ICall(callback(this, &Ctrl::WndDestroy0));
}

#ifdef PLATFORM_WIN32
void Ctrl::WndCreateCaret(const Rect& cr)
{
 ICall(THISBACK1(WndCreateCaret0, cr));
}
#endif

void Ctrl::WndShow(bool b)
{
 ICall(THISBACK1(WndShow0, b));
}

void Ctrl::WndUpdate()
{
 ICall(THISBACK(WndUpdate0));
}

void Ctrl::SetWndForeground()
{
 ICall(THISBACK(SetWndForeground0));
}

bool Ctrl::WndEnable(bool b)
{
 ICall(THISBACK1(WndEnable0, &b));
```

```
 return b;
}

bool Ctrl::SetWndFocus()
{
 bool b;
 ICall(THISBACK1(SetWndFocus0, &b));
 return b;
}

void Ctrl::WndInvalidateRect(const Rect& r)
{
 ICall(THISBACK1(WndInvalidateRect0, r));
}

void Ctrl::WndSetPos(const Rect& rect)
{
 ICall(THISBACK1(WndSetPos0, rect));
}

void Ctrl::WndUpdate(const Rect& r)
{
 ICall(THISBACK1(WndUpdate0r, r));
}

void  Ctrl::WndScrollView(const Rect& r, int dx, int dy)
{
 ICall(THISBACK3(WndScrollView0, r, dx, dy));
}

void Ctrl::EventLoop(Ctrl *ctrl)
{
 Call(callback1(&Ctrl::EventLoop0, ctrl));
}

END_UPP_NAMESPACE
```

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 09:34:28 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 29 January 2010 03:32
What is the NonMainMutex actually for? i think that is more or less the problem..

To block any other non-main thread trying to Call GUI thread at the same time.

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 10:17:31 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 29 January 2010 04:31one more:

indeed, in CtrlMT.cpp
void Ctrl::ICall(Callback cb)

is suffering a not release of semaphore due to an early CtrlCall.Clear in a DoCall somewhere in between, maybe from another thread??, so that the PerformCall cant be executed, which would release the semaphore...is there any other thread accessing DoCall?

This is exactly what NonGuiMutex is supposed to avoid... But something went wrong.

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 10:28:42 GMT
View Forum Message <> Reply to Message

Hm, try this please:

```
void Ctrl::ICall(Callback cb)
{
 LLOG("Ctrl::Call " << IsMainThread() << ", nonmain: " << NonMain);
 if(IsMainThread())
  cb();
 else {
  GuiLock __;
  CallBox cbox;
  cbox.cb = cb;
  CtrlCall = callback1(PerformCall, &cbox);
  int level = LeaveGMutexAll();
  WakeUpGuiThread();
  LLOG("Waiting for semaphore");
  if(!Thread::IsShutdownThreads())
   cbox.sem.Wait();
  EnterGMutex(level);
 }
```

```
 LLOG("-- Ctrl::Call " << IsMainThread());
}

void Ctrl::Call(Callback cb)
{
 if(IsMainThread())
  cb();
 else {
  GuiLock __;
  CallBox cbox;
  cbox.cb = cb;
  UPP::PostCallback(callback1(PerformCall, &cbox));
  int n = NonMain;
  int nn = n;
  NonMain = 0;
  for(int i = 0; i < n; i++)
   NonMainLock.Leave();
  int level = LeaveGMutexAll();
  WakeUpGuiThread();
  if(!Thread::IsShutdownThreads())
   cbox.sem.Wait();
  for(int i = 0; i < n; i++)
   NonMainLock.Enter();
  EnterGMutex(level);
  NonMain = n;
 }
}
```

(Rationale: It might be possible that Call gets invoked by different thread than the one that locked NonMainLock - in that case, setting NonMain to zero is indeed quite a bad idea).

All that convoluted mess just because M$ long time ago decided that HWND is per thread (not process)...

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Fri, 29 Jan 2010 10:46:04 GMT
View Forum Message <> Reply to Message

unfortunately no difference,

the custom CoWork still hangs. even faster , my app is still haniging in that..

isnt the problem that maybe another thread might bbe running the ProcessEvent, which race condition like will do a CtrlCall.Clear()?

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 10:57:57 GMT
View Forum Message <> Reply to Message

Another thread should not be running ProcessEvents, that is the whole point of this charade..

The real problem that causes all this is that GetMessage is only recieving messages for windows created by calling thread. That is why we need to "call" the main thread to both create windows and to recieve messages.

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 10:58:53 GMT
View Forum Message <> Reply to Message

Anyway, this should not hurt, please add and check:

```
bool Ctrl::ProcessEvent(bool *quit)
{
 ASSERT(IsMainThread());
```

Mirek

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Fri, 29 Jan 2010 11:30:09 GMT
View Forum Message <> Reply to Message

the ASSERT has not triggered, at least, i added also to DoCall, just to be sure. neither there.. this is really weired..

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 11:48:59 GMT
View Forum Message <> Reply to Message

OK, I think have a clue.

Try

```
Callback    Ctrl::CtrlCall;
static Mutex CtrlCallMutex;
```

```
void WakeUpGuiThread();

bool Ctrl::DoCall()
{
 LLOG("DoCall");
 GuiLock __;
 {
  Mutex::Lock __(CtrlCallMutex);
  CtrlCall();
  CtrlCall.Clear();
 }
 LLOG("--- DoCall, nonmain: " << NonMain);
 return NonMain;
}

void Ctrl::ICall(Callback cb)
{
 LLOG("Ctrl::Call " << IsMainThread() << ", nonmain: " << NonMain);
 if(IsMainThread())
  cb();
 else {
  GuiLock __;
  CallBox cbox;
  cbox.cb = cb;
  {
   Mutex::Lock __(CtrlCallMutex);
   CtrlCall = callback1(PerformCall, &cbox);
  }
  int level = LeaveGMutexAll();
  WakeUpGuiThread();
  LLOG("Waiting for semaphore");
  if(!Thread::IsShutdownThreads())
   cbox.sem.Wait();
  EnterGMutex(level);
 }
 LLOG("-- Ctrl::Call " << IsMainThread());
}
```

Rationale: ProcessEvents is possibly unlocked, I guess especially for DnD loop. It is possible there is a race condition between CtrlCall = assignment and CtrlCall.Clear (if execution of CtrlCall takes long enough).

Mirek

Subject: Re: DnD hangs in MT Refresh()ing

Posted by kohait00 on Fri, 29 Jan 2010 12:08:45 GMT
View Forum Message <> Reply to Message

GOOOD NEWS mirek...it works for me !! but not for the CoWork

i tested my software under pretty heavy conditions. my software does NOT hang. thanks a LOT!! that's driven me crazy a week now. i should have provided you the info that in deed, the worker threads perform a long enough task on the gui, just like you guessed. sorry for that. i think this will become a patch. are there any drawbacks on that? performance hits, deadlock potentials? (the more mutexes one uses

but the CoWork test has failed. it seems to have another problem in CoWork anyway, cause one can still click and see the title change the caption, so the message queue is runnning.

i could not grasp the exact error there, have you been able to reproduce the hang in the custom CoWork??

interestingly enough, when the test application "freezes", it seems only the worker threads freeze, as soon as you change the size of the window or minimize/maximize, it continues again.
just go on posting if you cant reproduce it. ill try it here.

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 12:21:21 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 29 January 2010 07:08GOOOD NEWS mirek...it works for me !! but not for the CoWork

i tested my software under pretty heavy conditions. my software does NOT hang. thanks a LOT!! that's driven me crazy a week now.

Glad to hear that. This stuff is really complicated...

Quote:
 i should have provided you the info that in deed, the worker threads perform a long enough task on the gui, just like you guessed. sorry for that. i think this will become a patch. are there any drawbacks on that? performance hits, deadlock potentials? (the more mutexes one uses

If there is any performance impact, it is negligible. And race conditions are bugs in any case! Incorrect code has to be fixed.

Quote:
but the CoWork test has failed. it seems to have another problem in CoWork anyway, cause one can still click and see the title change the caption, so the message queue is runnning.

i could not grasp the exact error there, have you been able to reproduce the hang in the custom CoWork??

Unfortunately, no. Frankly, this use of CoWork is very weird, but I understand it is just testcase.

In any case, the exit strategy of this testcase is invalid - it freezes on close and rightfully so. But that is another issue, I guess...

Quote:
interestingly enough, when the test application "freezes", it seems only the worker threads freeze, as soon as you change the size of the window or minimize/maximize, it continues again.
just go on posting if you cant reproduce it. ill try it here.

Yes, I cannot reproduce it...

Mirek

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Fri, 29 Jan 2010 12:32:55 GMT
View Forum Message <> Reply to Message

well indeed, it was only a test case, i was not worried about correct close or the like, it even produces mem leaks..
it was the easiest way to set up a bunch of worker threads that do something in the manner my software is doing (receiving stuff over sockets and completion port threads) and manipulating directly the gui.. but....it produced a different issue nevermind.
thank you a LOT really.

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 13:49:10 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 29 January 2010 07:32well indeed, it was only a test case, i was not worried about correct close or the like, it even produces mem leaks..
it was the easiest way to set up a bunch of worker threads that do something in the manner my software is doing (receiving stuff over sockets and completion port threads) and manipulating directly the gui.. but....it produced a different issue nevermind.
thank you a LOT really.

Thanks to you, stupid bug gone.

The problem with GuiLock is that it is relatively recent (2009) and it not many people are really

developing MT apps that depend on it. So some quirks are to be expected.

Still, we should try to fix a testcase as well... I will try on my dualcore notebook.

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Fri, 29 Jan 2010 15:40:54 GMT
View Forum Message <> Reply to Message

i will investigate my self on that as well this weekend, maybe i can provide something.

BTW: in heapdbg.cpp:60

```
static
#ifdef flagMT
#ifdef COMPILER_MSC
__declspec(thread)
#else
__thread
#endif
#endif
dword s_ignoreleaks;
```

could be replaced with

```
static thread__ dword s_ignoreleaks;
```

for consistency, so the thread__ def is used everywhere.

BTW2: remember the Timer package i postet recently? you asked to push it into bazaar and someone told me there to ask for bazzar svn commit rights..how is that? you also can put it there yourself.

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 29 Jan 2010 18:43:45 GMT
View Forum Message <> Reply to Message

ok...

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Mon, 01 Feb 2010 13:52:35 GMT
View Forum Message <> Reply to Message

hi mirek, just to be sure:

in Win32Wnd
in Create0 and ProcessEvent
you put and left a
ASSERT(IsMainThread());
..

is that really nesseccary? cause there might be cases where another thread starts to execute a dialog, or shouldnt that happen (because thats bad design, good would be gui = main thread)

cheers

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Tue, 02 Feb 2010 18:15:33 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Mon, 01 February 2010 08:52hi mirek, just to be sure:

in Win32Wnd
in Create0 and ProcessEvent
you put and left a
ASSERT(IsMainThread());
..

is that really nesseccary? cause there might be cases where another thread starts to execute a dialog, or shouldnt that happen (because thats bad design, good would be gui = main thread)

cheers

I believe it is ok - another thread can never execute a dialog DIRECTLY.

See explanation above - it is because all messages are associated with a thread that created the window (by Win32). That is the reason we need all that stupid machinery of "calling to the main thread" - if non-main thread wants to create a window, it is done by Call or ICall -> this puts the request for the main thread, which, when idle (event queue empty), adopts a request and performs it. Non-main thread waits blocked by semaphor, main thread releases the semaphor when request finished. That way, all windows are created by the main thread and all message loops are performed by the main thread too...

Mirek

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Wed, 03 Feb 2010 11:32:17 GMT
View Forum Message <> Reply to Message

Hey, I believe I have found why that CoWork test does not work...

It is SO twisted and obvious... Obviously, the main issue is that "Start" routine NEVER RETURNS (because it waits for all iterations to finish).

That means it effectively stops the timer queue...

Mirek

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Thu, 04 Feb 2010 11:37:35 GMT
View Forum Message <> Reply to Message

hi mirek,

i didn't quite get the point.. why should Start not return, it simply posts 10 tasks to the CoWork, which is not blocking normally..(not permanentl, besides accessing internals of CoWork protected by internal Critical Section).

the indicater for closinfg was the dtor of App() setting a flag, by that timethe application was dying, the threads could not comlete their Refresh.. thus not releasing GuiLock. and the dtor could not complete cause of CoWork dtor Finish() call. but why should the App dtor run in GuiLock mode??

nevertheless, here is a version using the same waiter Thread trick, recently used in my WorkQueueTest example..

this one should quit correctly..which might be a general approach of how to close/shutdown an MT application with other threads using GuiLock __;..

File Attachments
1) CoWork.rar, downloaded 444 times

Subject: Re: DnD hangs in MT Refresh()ing
Posted by mirek on Fri, 05 Feb 2010 09:43:54 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Thu, 04 February 2010 06:37hi mirek,

i didn't quite get the point.. why should Start not return, it simply posts 10 tasks to the CoWork, which is not blocking normally..(not permanentl, besides accessing internals of CoWork protected by internal Critical Section).

CoWork destructor waits for all tasks to finish.

Besides, not all of these 10 tasks are started in the process. CoWork is loop paralelizer, it starts only as much tasks (globally!) as there is cores in CPU (well, in fact, it starts 2 more to cover 'wait for I/O' issues, but that it is).

Mirek

Subject: Re: DnD hangs in MT Refresh()ing
Posted by kohait00 on Mon, 08 Feb 2010 20:55:24 GMT
View Forum Message <> Reply to Message

this multithreading stuff can drill a hole in the brain
thanks for help...
