

Re: corrupted pointer when initing a dll

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2007-05/msg00003.html>

- *From:* "Alexander Nickolov" <agnickolov@xxxxxxxx>
 - *Date:* Mon, 30 Apr 2007 17:35:20 -0700
-

Omitting irrelevant groups.

Your thread routine prototype is wrong. Make it:

```
// header
static DWORD WINAPI ParseA2IThread(LPVOID)

// cpp
DWORD WINAPI CCCP::ParseA2IThread(LPVOID) {

// Don't call _endthread – that can crash since you didn't use
_beginthread[ex]
return 0;
}
```

I didn't see where you set up your `m_strA2Lfname` – I leave it up to you to ensure it's set up correctly. Of course your class has to outlive its thread as well, so you need to wait for your thread in the class destructor or earlier. Note that closing a thread handle does not stop the thread – you essentially leak running threads in your code (not handles – the real threads). This most likely has no detectable consequences, but may cause a crash if your class is destroyed while its thread is still running. In an extreme case your class is housed in a DLL and that DLL can be unloaded while your thread is still running since its shut down completed sucessfully, but it didn't stop its threads. This is a guaranteed crash and when it happens and it's likely very hard to reproduce and diagnose.

With all of the above comments about gotchas, I still don't know where does your crash happen. Can you give me an exact place where your crash occurs? What does go wrong according to your debugging experience?

—

```
=====
Alexander Nickolov
Microsoft MVP [VC], MCSO
email: agnickolov@xxxxxxxx
MVP VC FAQ: http://vcfaq.mvps.org
```

=====
"jc" <k.jayachandran@xxxxxxxx> wrote in message
news:1177965742.942024.256360@xx

the corresponding code is

```
int CCCP::StartLoadA2lThread(void){
/*
function ccp thread
parameters: void
*/
//CParseWaitDialog *p_pwd;
DWORD dParseThreadID;
m_pWD = DisplayParseWaitDialog();
m_bA2lLoadThreadActive = TRUE;
if(m_hA2lLoadThread){
CloseHandle(m_hA2lLoadThread);
}
m_hA2lLoadThread = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)
ParseA2lThread, this, 0,&dParseThreadID);
if(m_hA2lLoadThread){
sGmsg.StringCopy("parsing a2l file thread started");
m_pParent->SendMessage(CM_PRINT_MESSAGE2);
}
else{
sGmsg.StringCopy("a2l parse thread not started");
m_pParent->SendMessage(CM_PRINT_MESSAGE2);
}
return 1;

}

void CCCP::ParseA2lThread(LPVOID p){
/*
*/
CCCP *w = (CCCP *)p;
w->LoadA2LDatabase(w->m_strA2Lfname);
_endthread();
}

int CCCP::LoadA2LDatabase(jcStr *fname){
/*
then call parser to parse the a2l file and load all the channel
values, if the parser returns with no error
return result of the load and parse.
zero is returned if no error in loading and parsing the a2l file
*/
int retval;

InitParser();//this will init the parser objects
char *temp;
//InitParser();
```

Re: corrupted pointer when initing a dll

```
//fname = m_strA2Lfname;
temp = fname->ReturnStr();
retval = Parser(temp, m_bA2lParseTalkative);
AnalyzeParseError(retval, fname, m_pWD);
DeleteParser();
InitTempLog();
m_pParent->PostMessage(CM_A2L_FILE_LOADED);
//AfterILoadA2lFile();
return retval;
}
thanks
jc
```

On Apr 30, 2:31 pm, "Alexander Nickolov" <agnicko...@xxxxxxx> wrote:

Make sure you are not passing pointers on your stack for your thread to use. Always allocate on the heap data you are to send to a different thread. Note this is just an educated guess as it's hard to tell what your problem is with no code posted...

--

=====
Alexander Nickolov
Microsoft MVP [VC], MCSD
email: agnicko...@xxxxxxx
MVP VC FAQ:<http://vcfaq.mvps.org>
=====

"jc" <k.jayachand...@xxxxxxxxxx> wrote in message

news:1177945430.435112.236610@xx

On Apr 30, 9:32 am, dasjotre <dasjo...@xxxxxxxxxxxxxxxx>
wrote:

On 30 Apr, 14:23, jc
<k.jayachand...@xxxxxxxxxx> wrote:

i'm developing a project
using vc++.
the main exe is a win32
application. it needs two
dlls. one is my
own
implementation of string
operations. the other dll is to
parse a2l
files(it is similar to xml
files, but it is based on asap2

Re: corrupted pointer when initing a dll

standard).
i created the parser using
bison and flex. then ported
the code to
vc+
+ environment including the
files lex.yy.c(output from
the flex
scanner)
a2lgrammar.tab.cpp(output
from bison) for post
processing
the
parser output and creating
the data structure. the result
is a dll
file that i link with my
application.

all was well till i realized
that this post processing the
parser
output(while the parser can
finish parsing the file in less
than a
minute for a typical a2l file
will be around 150,000
lines) takes
around 10 minutes, during
the entire application
freezes. so i
decided
to call the parsing using a
separate thread.

i call the parser with the file
name. before i call the
parser
function i initialize the
parser manually.

i reach the parser
application in two different
paths. in one path
there is a conf file, which

Re: corrupted pointer when initing a dll

has the info about the a2l
file and from
there it calls the parser
application. this goes well
with no
problem.

but if i try to load the a2l
file directly then the
filename pointer
gets corrupted. in different
ways
sometimes when i enter the
thread
sometimes when i initialize
the dll
sometimes when i parse the
file

but nothing happens when i
call the a2l parse and
process thread
after
i load the conf file.

i spent a week trying to
figure this out and realized
that this is
beyond me.
any help and pointer will be
appreciated

this is W32 not C++ problem

are you starting that thread inside
DLLMain?

check

this: http://boost.org/doc/html/threads/implementation_notes.html#threads.i...

Re: corrupted pointer when initing a dll

Hide quoted text –

– Show quoted text –

no i start the thread inside my application and call the dll functions from the thread.
i don't think i have anything like dllMain.
there are three functions that i use
0. start the thread (sometimes this file name pointer corrupts here)
1. initparser (this initialises the classes that parse, this is very useful for me because, this will let me reparse a different file without closing the application)(it corrupts here too)
2. parser(file name)(if it escaped before, now it is sure that it gets corrupted)
3. postprocess
4. copythe datastructure from the dll to the application(here i duplicate all the strucutre, as it will be deleted soon, but these data structure i need for the application)
4. deleteparser(delete all the memory that i allocated)

thanks
jc