Ryzom - Bug # 906

Status:	Closed	**Priority:**	Normal
Author:	rti	**Category:**	Client: General
Created:	05/15/2010	**Assignee:**	
Updated:	05/28/2010	**Due date:**	
Subject:	Ryzom Client - Custom libxml allocators and CXmlAutoPtr problem on Mac OS X (patch included)		

Description

Hi.

I am running Ryzom client on Mac OS 10.6 as x86_64.

What happens
Starting the client led to a crash (try to free() memory that was not allocated) in CInputHandlerManager::parseKey():

```
CXMLAutoPtr prop;
...
prop= (char*) xmlGetProp( cur, (xmlChar*) "id" );
...
prop= (char*) xmlGetProp( cur, (xmlChar*) "mod" ); // crash
...
```

The crash happens on the last line when CXmlAutoPtr::destroy() is called before assigning a newly allocated value to the auto pointer; it does:

```
if (_Value)
{
   xmlFree(const_cast<char *>(_Value));
   _Value = NULL;
}
```

The Ryzom client installs custom allocators for libxml in init.cpp.

```
xmlMemSetup (XmlFree4NeL, XmlMalloc4NeL, XmlRealloc4NeL, XmlStrdup4NeL);
```

The function XmlMalloc4NeL allocates a block like that:

```
-------------------------
|S|S|S|S|D|D|D|D|D|D|D|D|
-------------------------
        ^
```

Here 8 bytes of memory were requested |D|, the custom allocator prepends 4 bytes |S| to store the size of the allocated block, but returns a pointer to the first data byte |D|.

Ok, now the problem I observed:
CInputHandlerManager::parseKey() is part of the client binary, when calling xmlGetProp(), the xml allocators were already exchanged

in init.cpp and the pointer returned has 4 bytes prepended for size information. Now, CXmlAutoPointer::destroy() is called. This implementation is part of game_share shared library. For some reason, the xmlMemSetup() done in the "context" of the client binary had no effect on calls to libxml made from the game_share library. Therefor, the xmlFree() (in CXmlAutoPtr::destroy()) tries to deallocate the pure data part (using the stock xmlFree() function from libxml), which was allocated with 4 more bytes in front. So the deallocation fails.

Possible fix:
The attached patch simply moves the implementation of CXmlAutoPtr to xml_auto_ptr.h. This way, the implementation of CXmlAutoPtr::destroy() making the xmlFree() calls gets "locally" compiled into the game_share lib *and* the client binary "context". The result is, that destroy() calls made from the client binary use the custom xmlFree() logic installed by xmlMemorySetup(), and everything is fine. libxml calls made from game_share should still use the stock allocators.

I am not 100% sure what is happening here and why is it working this way. Might be somehow related to the linkage properties of the variables libxml is using to store the function pointers for xmlAlloc() / xmlFree() implementations. Any comments, theories, explanations welcome!

History

#1 - 05/15/2010 06:01 pm - vl

Another solution could be to just comment xmlMemSetup() and use default allocator. I don't really know the reason why they override it.

#2 - 05/15/2010 06:08 pm - rti

Did that too, but thought it would be too easy :)

When keeping the allocators, they should be changed to use size_t instead of the 32bit int, btw.
But then again this adds another 4 byte overhead to each allocation.

#3 - 05/17/2010 03:30 pm - vl

- Status changed from New to Resolved

- % Done changed from 0 to 100

Applied in changeset r180.

#4 - 05/18/2010 12:49 am - rti

I think this issue has to be reopened.

Now, as I am running in debug mode, a very similar error appears. But the other way around. A string allocated without the leading size information is being freed 4 bytes in front of it. I am still not sure what is really going on here. But seems like sometimes, the custom allocators for libxml kick it, sometimes not.

Disabling them completely works without any crashes.

client_mac_d(45068,0x7fff700c7be0) malloc: *** error for object 0x108eaa27c: pointer being freed was not allocated
Program received signal SIGABRT, Aborted.
0x00007fff80b8b886 in __kill ()
(gdb) bt

#0 0x00007fff80b8b886 in __kill ()
#1 0x00007fff80c2beae in abort ()
#2 0x00007fff80b43a75 in free ()
#3 0x0000000100032ef7 in __gnu_cxx::new_allocator<unsigned char>::deallocate (this=0x10104e458, __p=0x108eaa27c "acinPARTY") at new_allocator.h:97
#4 0x000000010031a61b in XmlFree4NeL (ptr=0x108eaa280) at /Users/rti/Development/ryzom/code/ryzom/client/src/init.cpp:184
#5 0x0000000100197c88 in CXMLAutoPtr::destroy (this=0x7fff5fbfe530) at xml_auto_ptr.h:61
#6 0x0000000100197cab in CXMLAutoPtr::~CXMLAutoPtr (this=0x7fff5fbfe530) at xml_auto_ptr.h:31
#7 0x0000000104965379 in CGenericXmlMsgHeaderManager::CNode::CNode (this=0x11cf23220, xmlNode=0x10997ab54, value=0) at /Users/rti/Development/ryzom/code/ryzom/common/src/game_share/generic_xml_msg_mngr.cpp:338
#8 0x00000001049650fe in CGenericXmlMsgHeaderManager::CNode::CNode (this=0x11cf23160, xmlNode=0x109978264, value=0) at /Users/rti/Development/ryzom/code/ryzom/common/src/game_share/generic_xml_msg_mngr.cpp:315
#9 0x00000001049650fe in CGenericXmlMsgHeaderManager::CNode::CNode (this=0x11cf230a0, xmlNode=0x10999c854, value=0) at /Users/rti/Development/ryzom/code/ryzom/common/src/game_share/generic_xml_msg_mngr.cpp:315
#10 0x0000000104966634 in CGenericXmlMsgHeaderManager::init (this=0x1010805a0, filename=@0x7fff5fbfee00) at /Users/rti/Development/ryzom/code/ryzom/common/src/game_share/generic_xml_msg_mngr.cpp:64
#11 0x000000010031d84f in postlogInit () at /Users/rti/Development/ryzom/code/ryzom/client/src/init.cpp:1178
#12 0x00000001001deb45 in main (argc=1, argv=0x7fff5fbff608) at /Users/rti/Development/ryzom/code/ryzom/client/src/client.cpp:520

#5 - 05/18/2010 10:22 am - vl

- Target version set to Version 0.8.0

#6 - 05/21/2010 11:13 am - rti

Same problem on Linux, see #922.

#7 - 05/21/2010 03:26 pm - rti

Seems like if the same issue occurred here again http://dev.ryzom.com/boards/17/topics/1936.

#8 - 05/22/2010 08:56 am - vl

We commented the function and code that manage the custom xml allocator. Let tell us if it works better or still crash.

#9 - 05/28/2010 05:24 pm - vl

- Status changed from Resolved to Closed

Files

ryzom_game_share_move_xml_auto_ptr_implementation_to_header.patch	16 kB	05/15/2010	rti