

Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Bug #1203

Build warning

February 12, 2023 16:32 - Artem S. Tashkinov

Status:	Rejected	Start date:	February 12, 2023
Priority:	Minor	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	4.3		
Affects version:	4.3		

Description

...  
Leaving directory aosd.  
In function ‘\_\_copy\_m’,  
inlined from ‘\_\_copy\_move\_a2’ at /usr/include/c++/12/bits/stl\_algobase.h:495:30,  
inlined from ‘\_\_copy\_move\_a1’ at /usr/include/c++/12/bits/stl\_algobase.h:522:42,  
inlined from ‘\_\_copy\_move\_a’ at /usr/include/c++/12/bits/stl\_algobase.h:529:31,  
inlined from ‘copy’ at /usr/include/c++/12/bits/stl\_algobase.h:620:7,  
inlined from ‘\_\_uninit\_copy’ at /usr/include/c++/12/bits/stl\_uninitialized.h:147:27,  
inlined from ‘uninitialized\_copy’ at /usr/include/c++/12/bits/stl\_uninitialized.h:185:15,  
inlined from ‘\_\_uninitialized\_copy\_a’ at /usr/include/c++/12/bits/stl\_uninitialized.h:372:37,  
inlined from ‘\_\_uninitialized\_move\_if\_noexcept\_a’ at /usr/include/c++/12/bits/stl\_uninitialized.h:397:2,  
inlined from ‘\_M\_fill\_insert’ at /usr/include/c++/12/bits/vector.tcc:594:7,  
inlined from ‘resize’ at /usr/include/c++/12/bits/stl\_vector.h:1032:18,  
inlined from ‘mc\_alloc’ at desmume/mc.cc:90:17,  
inlined from ‘MMU\_Init’ at desmume/MMU.cc:731:10,  
inlined from ‘NDS\_Init’ at desmume/NDSSystem.cc:43:10,  
inlined from ‘play’ at plugin.cc:334:17:  
/usr/include/c++/12/bits/stl\_algobase.h:431:30: warning: ‘\_\_builtin\_memcpy’ writing 1 or more bytes into a region of size 0 overflows  
the destination [-Wstringop-overflow=]  
431 | \_\_builtin\_memmove(\_result, \_\_first, sizeof(\_Tp) \* \_Num); | ^  
In member function ‘allocate’,  
inlined from ‘allocate’ at /usr/include/c++/12/bits/alloc\_traits.h:464:28,  
inlined from ‘\_M\_allocate’ at /usr/include/c++/12/bits/stl\_vector.h:378:33,  
inlined from ‘\_M\_fill\_insert’ at /usr/include/c++/12/bits/vector.tcc:575:45,  
inlined from ‘resize’ at /usr/include/c++/12/bits/stl\_vector.h:1032:18,  
inlined from ‘mc\_alloc’ at desmume/mc.cc:90:17,  
inlined from ‘MMU\_Init’ at desmume/MMU.cc:731:10,  
inlined from ‘NDS\_Init’ at desmume/NDSSystem.cc:43:10,  
inlined from ‘play’ at plugin.cc:334:17:  
/usr/include/c++/12/bits/new\_allocator.h:137:55: note: at offset 262144 into destination object of size 262144 allocated by ‘operator  
new’  
137 | return static\_cast<\_Tp\*>(\_GLIBCXX\_OPERATOR\_NEW(\_n \* sizeof(\_Tp))); |  
Successfully linked console.so.  
Leaving directory console.  
...  
  
No idea where it happens. Could it be console?

History

#1 - February 12, 2023 16:39 - Artem S. Tashkinov

This is when compiling/building audacious-plugins-4.3-beta1.tar.bz2

#2 - February 13, 2023 23:55 - Thomas Lange

Compile with -j1 or search for the "desmume/\*.cc" folder/file. You will see this warning is from linking xsf.  
--> <https://github.com/audacious-media-player/audacious-plugins/pull/106#issuecomment-1133286466>

Which compiler and build system do you use? Does the workaround work for you too?

### #3 - February 21, 2023 07:48 - Artem S. Tashkinov

Thomas Lange wrote:

Compile with -j1 or search for the "desmume/\*.cc" folder/file. You will see this warning is from linking xsf.  
--> <https://github.com/audacious-media-player/audacious-plugins/pull/106#issuecomment-1133286466>

Which compiler and build system do you use? Does the workaround work for you too?

I use Fedora 37.

```
gcc --version
gcc (GCC) 12.2.1 20221121 (Red Hat 12.2.1-4)

export CFLAGS="-O2 -mtune=generic -pipe -flto=auto"
export CXXFLAGS="$CFLAGS"
```

The issue is here:

```
Successfully compiled desmume/slot1_retail.cc (plugin).
(the same errors)
Successfully linked xsf.so.
```

### #4 - February 21, 2023 07:50 - Artem S. Tashkinov

The linked patch fixes the issue.

```
diff --git a/src/xsf/desmume/mc.cc b/src/xsf/desmume/mc.cc
index dd6e670b1..648a5c89e 100644
--- a/src/xsf/desmume/mc.cc
+++ b/src/xsf/desmume/mc.cc
@@ -87,7 +87,7 @@ void mc_init(memory_chip_t *mc, int type)
    uint8_t *mc_alloc(memory_chip_t *mc, uint32_t size)
    {
        mc->data.clear();
-       mc->data.resize(size, 0);
+       mc->data.resize(size);
        mc->size = size;
        mc->writeable_buffer = true;
```

**#5 - February 21, 2023 17:27 - John Lindgren**

This seems to be the same as [https://gcc.gnu.org/bugzilla/show\\_bug.cgi?id=100366](https://gcc.gnu.org/bugzilla/show_bug.cgi?id=100366) which will supposedly be fixed in GCC 12.3.

**#6 - February 21, 2023 18:05 - John Lindgren**

- *Status changed from New to Rejected*

Closing, as it's a compiler bug. The warning is a false-positive and harmless.