

Angular 7/8 FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory #13734



arrayik-yervandyan opened this issue on Feb 21 · 167 comments

Assignees



Labels

comp: devkit/build-angular

freq3: high

severity4: memory/performance

type: bug/fix

Milestone

8.1.x



arrayik-yervandyan commented on Feb 21 • edited by alan-agius4 ▾



Bug report

Command (mark with an x)

```
- [ ] new
- [X] build
- [ ] serve
- [ ] test
- [ ] e2e
- [ ] generate
- [ ] add
- [ ] update
- [ ] lint
- [ ] xi18n
- [ ] run
- [ ] config
- [ ] help
```

Description

With the last merge project build process is not working. I get failure that is pasted below.
Tried to change --max-old-space-size=4096 still not working.
Any suggestions what this can be ?



Minimal Reproduction

run command `ng build --prod`
in angular.json file we have

```
"production": {
  "optimization": true,
  "outputHashing": "all",
  "sourceMap": false,
  "extractCss": true,
  "namedChunks": false,
  "aot": true,
  "extractLicenses": true,
  "vendorChunk": false,
  "buildOptimizer": true,
  "fileReplacements": [
    {
      "replace": "src/environments/environment.ts",
      "with": "src/environments/environment.prod.ts"
    }
  ]
},
```



Exception or Error

Here is the log file.

```

1: 0x00000000 Node::Abort() [ng build --prod --configuration=config --output-path=test --base-href=/test/]
2: 0x8dbaec [ng build --prod --configuration=config --output-path=test --base-href=/test/]
3: 0xad83de v8::Utils::ReportOOMFailure(v8::internal::Isolate*, char const*, bool) [ng build --prod --configuration=config --
output-path=test --base-href=/test/]
4: 0xad8614 v8::internal::V8::FatalProcessOutOfMemory(v8::internal::Isolate*, char const*, bool) [ng build --prod --
configuration=config --output-path=test --base-href=/test/]
5: 0xec5c42 [ng build --prod --configuration=config --output-path=test --base-href=/test/]
6: 0xec5d48 v8::internal::Heap::CheckIneffectiveMarkCompact(unsigned long, double) [ng build --prod --configuration=config --
output-path=test --base-href=/test/]
7: 0xed1e22 v8::internal::Heap::PerformGarbageCollection(v8::internal::GarbageCollector, v8::GCCallbackFlags) [ng build --prod
--configuration=config --output-path=test --base-href=/test/]
8: 0xed2754 v8::internal::Heap::CollectGarbage(v8::internal::AllocationSpace, v8::internal::GarbageCollectionReason,
v8::GCCallbackFlags) [ng build --prod --configuration=config --output-path=test --base-href=/test/]
9: 0xed53c1 v8::internal::Heap::AllocateRawWithRetryOrFail(int, v8::internal::AllocationSpace,
v8::internal::AllocationAlignment) [ng build --prod --configuration=config --output-path=test --base-href=/test/]
10: 0xe9d636 [ng build --prod --configuration=config --output-path=test --base-href=/test/]
11: 0xeafee7 v8::internal::Factory::NewLoadHandler(int) [ng build --prod --configuration=config --output-path=test --base-
href=/test/]
12: 0xf2f4db v8::internal::LoadHandler::LoadFromPrototype(v8::internal::Isolate*, v8::internal::Handle<v8::internal::Map>,
v8::internal::Handle<v8::internal::JSReceiver>, v8::internal::Handle<v8::internal::Smi>,
v8::internal::MaybeHandle<v8::internal::Object>, v8::internal::MaybeHandle<v8::internal::Object>) [ng build --prod --
configuration=config --output-path=test --base-href=/test/]
13: 0xf36d1f v8::internal::LoadIC::ComputeHandler(v8::internal::LookupIterator*) [ng build --prod --configuration=config --
output-path=test --base-href=/test/]
14: 0xf3d94c v8::internal::LoadIC::UpdateCaches(v8::internal::LookupIterator*) [ng build --prod --configuration=config --
output-path=test --base-href=/test/]
15: 0xf3dffc v8::internal::LoadIC::Load(v8::internal::Handle<v8::internal::Object>, v8::internal::Handle<v8::internal::Name>)
[ng build --prod --configuration=config --output-path=test --base-href=/test/]
16: 0xf42935 v8::internal::Runtime_LoadIC_Miss(int, v8::internal::Object**, v8::internal::Isolate*) [ng build --prod --
configuration=config --output-path=test --base-href=/test/]
17: 0x235e3925be1d

```



Your Environment

Angular CLI: 7.3.1
Node: 10.15.1

... heap; platform-browser; platform-browser-dynamic; router

Package	Version

@angular-devkit/architect	0.13.2
@angular-devkit/build-angular	0.13.2
@angular-devkit/build-optimizer	0.13.2
@angular-devkit/build-webpack	0.13.2
@angular-devkit/core	7.3.2
@angular-devkit/schematics	7.3.1
@angular/cdk	7.3.3
@angular/cli	7.3.1
@ngtools/webpack	7.3.2
@schematics/angular	7.3.1
@schematics/update	0.13.1
rxjs	6.4.0
typescript	3.1.6
webpack	4.29.0

 104

 8





alan-agius4 commented on Feb 21

Collaborator

Hi, in the error stack trace, it seems that you are using a different configuration from "production", can you kindly share that configuration?



arayik-yervandyan commented on Feb 21 • edited ▾

Author

Sorry for mismatch here is configuration params that you need

```
    "sourceMap": false,  
    "extractCss": true,  
    "namedChunks": false,  
    "aot": true,  
    "extractLicenses": true,  
    "vendorChunk": false,  
    "buildOptimizer": true,  
    "fileReplacements": [  
      {  
        "replace": "src/environments/environment.ts",  
        "with": "src/environments/environment.config.ts"  
      }  
    ]  
  }  
}
```



  **alan-agius4** added the `severity4: memory/performance` label on Feb 25



nicobytes commented on Feb 26 • edited ▼

I have the same error:

```
3: 0xab730e v8::Utils::ReportOOMFailure(v8::internal::Isolate*, char const*, bool) [ng build --prod --
configuration=stag --source-map]
4: 0xab7528 v8::internal::V8::FatalProcessOutOfMemory(v8::internal::Isolate*, char const*, bool) [ng build --prod --
configuration=stag --source-map]
5: 0xea5152 [ng build --prod --configuration=stag --source-map]
6: 0xeb10aa v8::internal::Heap::PerformGarbageCollection(v8::internal::GarbageCollector, v8::GCCallbackFlags) [ng
build --prod --configuration=stag --source-map]
7: 0xeb1a14 v8::internal::Heap::CollectGarbage(v8::internal::AllocationSpace, v8::internal::GarbageCollectionReason,
v8::GCCallbackFlags) [ng build --prod --configuration=stag --source-map]
8: 0xeb4345 v8::internal::Heap::AllocateRawWithRetry(int, v8::internal::AllocationSpace,
v8::internal::AllocationAlignment) [ng build --prod --configuration=stag --source-map]
9: 0xe7c824 v8::internal::Factory::NewFillerObject(int, bool, v8::internal::AllocationSpace) [ng build --prod --
configuration=stag --source-map]
10: 0x111e1de v8::internal::Runtime_AllocateInNewSpace(int, v8::internal::Object**, v8::internal::Isolate*) [ng build -
--prod --configuration=stag --source-map]
11: 0x35a4176041bd
Aborted (core dumped)
npm ERR! code ELIFECYCLE
npm ERR! errno 134
npm ERR! senseta-dashboard@1.2.0 build: `ng build "--prod" "--configuration=stag" "--source-map"`
npm ERR! Exit status 134
npm ERR!
npm ERR! Failed at the senseta-dashboard@1.2.0 build script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!     /root/.npm/_logs/2019-02-26T16_41_37_961Z-debug.log
```

My env:



Bloodcast69 commented on Feb 27

Guys here's described the "workaround" for this problem: [#5618](#)



1



mumairofficial commented on Mar 18 • edited ▼

- ☐ new
- ☐ build
- ☒ serve prod
- ☐ test
- ☐ e2e

I am facing same issue while `ng serve --prod`


```

locally. It hasn't been reviewed for security issues.

Binding this server to an open connection can result in compromising your application or
computer. Using a different host than the one passed to the "--host" flag might result in
websocket connection issues. You might need to use "--disableHostCheck" if that's the
case.
*****
This is a simple server for use in testing or debugging Angular applications locally.
It hasn't been reviewed for security issues.

DON'T USE IT FOR PRODUCTION!
*****
** Angular Live Development Server is listening on m3-local.mindsphereapps.io:4200, open your browser on http://m3-local.mindsph
ereapps.io:4200/ **
 92% chunk asset optimization TerserPlugin
<--- Last few GCs --->

[2016:000002148560E200] 374847 ms: Scavenge 1344.5 (1423.7) -> 1343.8 (1424.2) MB, 4.5 / 0.0 ms (average mu = 0.177, current
mu = 0.104) allocation failure
[2016:000002148560E200] 374855 ms: Scavenge 1344.7 (1424.2) -> 1344.0 (1424.7) MB, 4.4 / 0.0 ms (average mu = 0.177, current
mu = 0.104) allocation failure
[2016:000002148560E200] 374862 ms: Scavenge 1344.8 (1424.7) -> 1344.2 (1425.7) MB, 4.7 / 0.0 ms (average mu = 0.177, current
mu = 0.104) allocation failure

<--- JS stacktrace --->

==== JS stack trace =====
    0: ExitFrame [pc: 000001658965C5C1]
Security context: 0x027a8189e6e1 <JSObject>
    1: SourceNode_walk [0000035134C44E79] [C:\Users\makbar\Desktop\code\rrr-portals\node_modules\webpack-sources\node_modules\so
urce-map\lib\source-node.js:~221] [pc=000001658E26CB47](this=0x00294dab0089 <SourceNode map = 0000039DEFD666F9>,aFn=0x01b1dcc345
81 <JSFunction (sfi = 000002B21DE70F71)>)
    2: SourceNode_walk [0000035134C44E79] [C:\Users\makbar\D...

FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory
 1: 00007FF7EBFBECF5
 2: 00007FF7EBF981A6
 3: 00007FF7EBF98BB0
 4: 00007FF7EC229AAE
 5: 00007FF7EC2299DF
 6: 00007FF7EC767724
 7: 00007FF7EC75DE87
 8: 00007FF7EC75C3FC
 9: 00007FF7EC765377
10: 00007FF7EC7653F6
11: 00007FF7EC3084B7
12: 00007FF7EC3A019A
13: 000001658965C5C1

```



I to got the same error. NodeJs process uses around 1.5GB of RAM, and after few minutes of attempt to clean/allocate memory, it fails with error.

```
$ ng serve --prod
*****
This is a simple server for use in testing or debugging Angular applications locally.
It hasn't been reviewed for security issues.

DON'T USE IT FOR PRODUCTION!
*****
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory
 1: 00007FF701C9ACB5
 2: 00007FF701C744E6
 3: 00007FF701C74EF0
 4: 00007FF701F047FE
 5: 00007FF701F0472F
 6: 00007FF702438964
 7: 00007FF70242F0F2
 8: 00007FF70242D68C
 9: 00007FF7024365B7
10: 00007FF702436636
11: 00007FF701FE1677
12: 00007FF7020789D2
13: 00000376121DC5C1
```



 themanojshukla commented on Mar 26

UPDATE :-

I've seen this issue only on **WINDOWS** machine particularly. Anyone else, noticed that? Or have got same issue in other machines?

Note: I tried 2 Windows machine, all tests gave same result. Both machine have same config almost.
i.e. 8GB RAM/Admin Access For Bash/CMD/PowerShell/ i3 & i5 Processor (quad core, both).

I tried MESSY project's code, it never compile on default node heap allocation (around 1.2 GB, not sure.). It works if I allocate more heap memory to node process, say 5 GB. But, in that case too, it took more than 30 mins to build. I tried the 5 GB node heap option with and without `--aot` AND/OR `--prod` .

Same code when tried on centos could instance of 1GB RAM/ Virtual Xenon Processor (1 Core), generates build within 90 Seconds. with and without `--aot` AND/OR `--prod`

I'm not sure, but, looks like problem is with **Node Service** not with Angular CLI itself, coz I've seen some performance lag in new NodeJS releases after last update (locally on WINDOWS machine only).

Some observations below:-

Even though allocating 5 GB of heap, it never used more than 1.9 GB (max) and allocation process was very slow.. it was increasing RAM consumption 1-2 MB per 3-5 seconds. And, was not using Disk Resource as it used to be in older version. (Disk access was very high previously on WINDOWS machine, but build process was faster without extra RAM allocation. This time, same machine almost same code with some bug fixes, but node was upgraded, disk consumption was lowered but slow build generation.)

Hope somebody can confirm this.



2



15



tibinthomas commented on Mar 27

Yes, I am also facing the same issue



2



sswayne commented on Mar 28

Same issue here running on a Mac book pro. macOS Mojave. Any environment with production: true

 **nokhodian** commented on Mar 28 • edited ▼

same issue on ubuntu server 18.04, but this solved my problem:

```
node --max_old_space_size=8192 node_modules/@angular/cli/bin/ng build --prod
```



 **themanojshukla** commented on Mar 28 • edited ▼

I've tried uninstalling Angular CLI 7, and installed 6.0.8. verified the downgraded version by `ng version`, still it breaks with the same error.

Maybe this could be because of nodeJs too. Coz, I'm building same code in CentOS 7, 1.75GB RAM (with 512MB swap memory), single core processor, and with angular CLI 7, it works there. It is breaking in my 2 windows machines.

For those, who can't afford allocating more RAM (due to less memory available) can create a SWAP memory from the space available from Hard Drive of system. It'll be little slower, but should work without System Upgrade. I've done the same. 👍



 **j2L4e** commented on Apr 5

In my case the problem only appears, when building `--prod --source-map`. The normal `--prod` build runs just fine.

Using `--max_old_space_size=8192` as suggested by [@nokhodian](#) it works with memory consumption peaking at ~3.8GiB during source map generation.

Build environment:

OS: Linux x86_64

Angular: 7.2.12

... animations, common, compiler, compiler-cli, core, forms

... http, language-service, platform-browser

... platform-browser-dynamic, router, service-worker

Package	Version

@angular-devkit/architect	0.10.4
@angular-devkit/build-angular	0.10.4
@angular-devkit/build-optimizer	0.10.4
@angular-devkit/build-webpack	0.10.4
@angular-devkit/core	7.0.4
@angular-devkit/schematics	7.1.0
@angular/cdk	7.3.7
@angular/cli	7.3.8
@angular/flex-layout	7.0.0-beta.19
@angular/material	7.3.7
@angular/pwa	0.11.0
@ngtools/webpack	7.0.4
@schematics/angular	7.1.0
@schematics/update	0.13.8
rxjs	6.4.0
typescript	3.2.4
webpack	4.19.1



5



arrayik-yervandyan commented on Apr 10

Author

In my angular.json file I have setted "sourceMap": false, but it doesn't helped me. Same issue for prod build



just chiming in here with same error message when running `ng build` when the build is working on a particular scss file



geo242 commented on Apr 18

I also have this issue but can resolve it with "`NODE_OPTIONS=--max-old-space-size=8192`". Ram usage during build gets up about 14GB.



2



ormico commented on Apr 18

Same here. I found that after further searching. I've sent out instructions to our team to set this environment variable and set it on our build servers. Seems to work and build seems to be faster than before it stopped working.



Calidus commented on Apr 18

I started running into the same issue with Azure Pipelines: [#microsoft/azure-pipelines-image-generation#854](#) I assumed it was a hosting problem, since don't have the issue on premise.



j3gb3rt commented on Apr 19

```
"old_space": {  
  "memorySize": 1280512000,  
  "committedMemory": 1278567272,  
  "capacity": 1155253856,  
  "used": 1152333192,  
  "available": 2920664  
},
```

Above is my old_space head info from the generated report. What could be making these numbers get so large?



j3gb3rt commented on Apr 19

Has anyone experienced this in a project that doesn't use sass?



5



hvdscbeer commented on Apr 25

We are experiencing this problem when we use



```
ng build --prod
```

Sometimes needed because we use different environment settings.



```
node --max_old_space_size=8192 node_modules/@angular/cli/bin/ng serve --prod
```

solves the problem but the build time is exceeded to 3 minutes.

I don't think this issue is solved

  **filipesilva** added the `comp: devkit/build-angular` label on Apr 26

  **ngbot** `bot` added this to the `needsTriage` milestone on Apr 26

  **filipesilva** added the `need: repro steps` label on Apr 26

 **filipesilva** commented on Apr 26

Member

This issue is the same as [#5618 \(comment\)](#), as far as I can tell.

I don't see a lot of things I can follow up and investigate though. [@themanojshukla](#) mentioned Windows was [taking a lot longer](#). [@j3gb3rt](#) is asking about sass, and I have seen reports in the past about sass making things slower.

Does anyone have a reproduction that we can look at and try to debug?

 1 

 **Calidus** commented on Apr 26

I have run into this issue on a project that does not use sass. I did some environment testing because I originally thought it was an issue relating to Azure Dev Ops.

Using Azure Dev Ops Pipelines:

- Hosted Linux agent with Node 10: Build succeeds
- Hosted Windows vs2017 Agent with Node 10: Build fails

• Hosted windows vs2019 Agent with Node 6: Build fails

- Hosted Windows Container Agent with Node 10: Build Failed 5/10 attempts
- Install Build Agent on my dev machine(win10) with Node 10: Build Succeeds
- Install Build Agent on my dev machine(win10) with Node 6: Build Succeeds
- Dev machine(win10) using cmd with Node 10: Build Succeeds
- Dev machine(win10) using cmd with Node 6: Build Succeeds



2



 **themanojshukla** commented on Apr 26

Yes @filipesilva and @Calidus my projects also don't use sass.

Let me give some more observations...

- The pet project (following best practices as much as can be) I created in around a year ago was working very fine that time (I don't remember the CLI version I used). Now when I'm building it using `--aot --prod` it takes bit more time than before but never breaks. No sass used.
- New project (of course production code base & I admit, really really messy codes), it breaks on `--aot --prod` . No Sass used. Breaks on windows always. Sometimes works on ubuntu without allocating extra heap on node process, but success rate on ubuntu is 10% say.
- Another mid-level project (messy codes), works without extra heap allocation but takes a bit more time on both **Windows & Ubuntu**.
- Another pet project (following best practices as much as can be) works fine so far.. but very less files to compile, so can't be sure on processing loads.

When running `ng build --aot --prod` for the first time after machine start/restart. it takes 57seconds. afterwards, 22-25seconds only.

```
Time: 56798ms
chunk {6} 6.5513600d892911c0b5c8.chunk.js () 10.5 kB [rendered]
chunk {scripts} scripts.0b0209148b202946cb76.bundle.js (scripts) 126 kB [initial] [rendered]
chunk {0} 0.e39df6c75a91500881b6.chunk.js (common) 3.18 kB [rendered]
chunk {1} 1.2a1024a32cabd50010a4.chunk.js () 40.6 kB [rendered]
chunk {2} 2.72a2c07aace60af88237.chunk.js () 11.8 kB [rendered]
chunk {3} 3.e9f5350cac79264aef94.chunk.js () 25.5 kB [rendered]
chunk {4} 4.a3ec448317ea334c0c5e.chunk.js () 17 kB [rendered]
chunk {5} 5.18ce233859e65a817ccb.chunk.js () 1.07 kB [rendered]
chunk {7} main.b63233f30e2bdda4874b.bundle.js (main) 490 kB [initial] [rendered]
chunk {8} polyfills.2d715b1323ab28f9b93d.bundle.js (polyfills) 42.3 kB [initial] [rendered]
chunk {9} styles.d66c33fdc2ced5111a89.bundle.css (styles) 164 kB [initial] [rendered]
chunk {10} inline.7add196a826c8f28a709.bundle.js (inline) 1.55 kB [entry] [rendered]
PS C:\AppData>
PS C:\AppData>
```

Second Run Screenshot..

```
Date: 2019-04-26T14:38:12.835Z
Hash: 0bc57a67498a27e6b001
Time: 22917ms
chunk {6} 6.5513600d892911c0b5c8.chunk.js () 10.5 kB [rendered]
chunk {scripts} scripts.0b0209148b202946cb76.bundle.js (scripts) 126 kB [initial] [rendered]
chunk {0} 0.e39df6c75a91500881b6.chunk.js (common) 3.18 kB [rendered]
chunk {1} 1.2a1024a32cabd50010a4.chunk.js () 40.6 kB [rendered]
chunk {2} 2.72a2c07aace60af88237.chunk.js () 11.8 kB [rendered]
chunk {3} 3.e9f5350cac79264aef94.chunk.js () 25.5 kB [rendered]
chunk {4} 4.a3ec448317ea334c0c5e.chunk.js () 17 kB [rendered]
chunk {5} 5.18ce233859e65a817ccb.chunk.js () 1.07 kB [rendered]
chunk {7} main.b63233f30e2bdda4874b.bundle.js (main) 490 kB [initial] [rendered]
chunk {8} polyfills.2d715b1323ab28f9b93d.bundle.js (polyfills) 42.3 kB [initial] [rendered]
chunk {9} styles.d66c33fdc2ced5111a89.bundle.css (styles) 164 kB [initial] [rendered]
chunk {10} inline.7add196a826c8f28a709.bundle.js (inline) 1.55 kB [entry] [rendered]
PS C:\AppData> █
```

Both are for same project.. and are of while writing this comment.

Hope it helps.



noopur-dabhi commented on May 2

I'm facing the same issue. Every time I build I get "JavaScript heap out of memory". I don't want to use `node --max_old_space_size=8192` since this doesn't sound like best practice. Any other way to fix this?



themanojshukla commented on May 3

@**noopur-dabhi** I don't think it is fair to say that using `node --max_old_space_size=8192` is not best practice.

It is nothing but allowing NODE process (*which is going to build your app now*) to use more than DEFAULT heap memory (from your RAM or SWAP area).

And this type of allocation is very very common in most of the app development (especially deployment) process. Example, in Java, if we want to restrict our java app heap usage, we add additional flag like `--Xmx64m` this means maximum heap memory allowed is 64MB.

So if 'Best Practice' is the only concern, I think it's not a problem using `node --max_old_space=8192`, this means you are allowing NODE process to use maximum of 8192MB from memory. nothing to do with Best Practice Concern.



7



3



lcpautotester commented on May 7 • edited ▼

This is still crashing for me> any suggestions?

I tried `"ng build --aot --prod"`

and `"node --max_old_space_size=8192 node_modules/@angular/cli/bin/ng build --prod"`

[9568:000002A08DC547B0] 183499 ms: Mark-sweep 1350.2 (1423.6) -> 1350.2 (1424.1) MB, 952.0 / 0.0 ms (average mu = 0.111, current mu = 0.000) allocation failure GC in old space requested

[9568:000002A08DC547B0] 184477 ms: Mark-sweep 1350.2 (1424.1) -> 1350.2 (1424.1) MB, 977.5 / 0.0 ms (average mu = 0.058, current mu = 0.000) last resort GC in old space requested

<--- JS stacktrace --->

==== JS stack trace =====

0: ExitFrame [pc: 000001D7B1850481]

Security context: 0x008d8961d9b1

1: /* anonymous */ [0000006BEE99B319] [C:\CODE\V2\UX\node_modules\webpack-sources\node_modules\source-map\lib\source-node.js:~342] [pc=000001D7B3E70B48](this=0x01dae2403e31,0x0263a30a1241 <String[6]: "Mill">,0x030059303561)

2: SourceNode_walk [000001779ECCDF31] [C:\CODE\V2\UX\node_modules\webpack-so...

FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory

```
2: 00007FF775DC8956 uv_loop_fork+85542
3: 00007FF775DC941D uv_loop_fork+88301
4: 00007FF7761EE72E void __cdecl v8::internal::FatalProcessOutOfMemory(class v8::internal::Isolate * __ptr64,char const * __ptr64)+798
5: 00007FF7761EE667 void __cdecl v8::internal::FatalProcessOutOfMemory(class v8::internal::Isolate * __ptr64,char const * __ptr64)+599
6: 00007FF7762A2144 public: static bool __cdecl v8::internal::Heap::RootIsImmortalImmovable(int)+14068
7: 00007FF776297F52 public: bool __cdecl v8::internal::Heap::CollectGarbage(enum v8::internal::AllocationSpace,enum
v8::internal::GarbageCollectionReason,enum v8::GCCallbackFlags) __ptr64+7234
8: 00007FF776296768 public: bool __cdecl v8::internal::Heap::CollectGarbage(enum v8::internal::AllocationSpace,enum
v8::internal::GarbageCollectionReason,enum v8::GCCallbackFlags) __ptr64+1112
9: 00007FF776295C25 public: void __cdecl v8::internal::Heap::CollectAllGarbage(int,enum v8::internal::GarbageCollectionReason,enum
v8::GCCallbackFlags) __ptr64+949
10: 00007FF7762A0174 public: static bool __cdecl v8::internal::Heap::RootIsImmortalImmovable(int)+5924
11: 00007FF7763D7E4D private: class v8::internal::HeapObject * __ptr64 __cdecl v8::internal::Factory::AllocateRawArray(int,enum
v8::internal::PretenureFlag) __ptr64+61
12: 00007FF7763D87E2 private: class v8::internal::Handle __cdecl v8::internal::Factory::NewFixedArrayWithFiller(enum
v8::internal::Heap::RootListIndex,int,class v8::internal::Object * __ptr64,enum v8::internal::PretenureFlag) __ptr64+66
13: 00007FF77641B0BA public: static class v8::internal::wasm::AsmType * __ptr64 __cdecl v8::internal::wasm::AsmType::Void(void)+69050
14: 00007FF776407A06 class v8::internal::MaybeHandle __cdecl v8::internal::BigIntLiteral(class v8::internal::Isolate * __ptr64,char const *
__ptr64)+127094
15: 00007FF77649AFA0 public: static int __cdecl v8::internal::StoreBuffer::StoreBufferOverflow(class v8::internal::Isolate * __ptr64)+64256
16: 000001D7B1850481
npm ERR! code ELIFECYCLE
npm ERR! errno 134
npm ERR! Exit status 134
npm ERR!
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
```



 **tonyholt** commented on Sep 14 • edited ▼

Wanted to add a bit more of information. ng build seems to work fine. only when using the --prod flag is where the heap issue happens. Here's my prod configuration from the angular.json:

```
"production": {  
  "fileReplacements": [{  
    "replace": "apps/standard/src/environments/environment.ts",  
    "with": "apps/standard/src/environments/environment.prod.ts"  
  }],  
  "optimization": true,  
  "outputHashing": "all",  
  "sourceMap": false,  
  "extractCss": true,  
  "namedChunks": false,  
  "aot": true,  
  "extractLicenses": true,  
  "vendorChunk": false,  
  "buildOptimizer": true,  
  "vendorSourceMap": false  
},
```



 **EkiGui** commented on Sep 16 • edited ▼

Thanks a lot @filipesilva for the exhaustive benchmark. Installing node-sass didn't pull us out but upgrading angular to v8.2.6 did the trick.



2

**akvaliya** commented on Sep 16

Is there any way to resolve this issue without updating node to current from LTS? In my project build gets generated in server. I can't ask all servers to update node.

**tomgruszowski** commented on Sep 16

@akvaliya node 12 will be LTS in less than a month. I can't see how a bug this intricate & wide reaching makes sense to tackle using a version that's about to become 'semi-legacy'. <https://nodejs.org/en/about/releases/>

**akvaliya** commented on Sep 16

@tomgruszowski Yes. But i can't tell client to update all his servers.

**benelliott** commented on Sep 16

@akvaliya Why are your application servers building the front-end?

 **akvaliya** commented on Sep 17

@benelliott Well i just push code to repo & testers from client team will pull & redeploy with build.

I don't this should be the point that why are we building fron-ent in server. Some client wants code to auto redeploy after commit. Like using jenkins.



 **d1sco** commented on Sep 20 • edited ▼

for me the solution was editing my tsconfig (ionic4 and angular8)

i was playing around adding properties to tsconfig, towards the end i enabled source maps for my prod build inside `angular.json` .

i had to remove the `sourceMap: true` and `sourceBase: "/"` from tsconfig , after that `--prod` works like a charm



 **kumaresan-subramani** commented on Sep 24

```
"scripts": {  
  "start": "node --max-old-space-size=4096 ./node_modules/@angular/cli/bin/ng serve",  
  "start:prod": "node --max-old-space-size=4096 ./node_modules/@angular/cli/bin/ng serve --prod",  
}
```

Above config solves for me



StevenSerrien commented on Sep 26 • edited ▼

For us upgrading from Angular 8.2.0 to 8.2.8 did the trick and we no longer got the error. While allocating more memory did not work for us.



irux commented on Sep 30

| @filipesilva just install Node 12.8 solved the issue . thanks.

I have the same issue with 12.8 too.



mindte commented on Oct 2

| @filipesilva just install Node 12.8 solved the issue . thanks.

Was getting with debug and Node >12.8 fix for me



trusktr commented on Oct 2

I have a custom project not using Angular but built with Webpack, and getting this error when I build with Webpack in `production` mode. I tried `--max_old_space_size=8192` but no luck (I guess I need more).



Dev4mir commented on Oct 2

I had the same issue and i did these steps:

- `ng update @angular/cli @angular/core`
 - update node to v12.11.1
- works fine with `--max_old_space_size=8192`
"@angular/core": "8.2.8", "@angular/cli": "^8.3.6"



Splaktar referenced this issue on Oct 3

refactor: convert app to latest Angular CLI format #641

📋 2 of 2 tasks complete

Merged



trusktr commented on Oct 3

The specific cause of this error for me (in my non-angular project) is the use of `uglifyjs-webpack-plugin`. It uses a ton a memory. For now I disabled minification in production, and will have to figure out how to fix it later.



Splaktar commented on Oct 3 • edited ▼

Member

I disabled minification in production

```
"build:highmem": "node --max_old_space_size=8192 ./node_modules/@angular/cli/bin/ng build",
```

I only started to need this after updating Angular and CLI from 8.0.0 to the latest 8.2.8 and 8.3.6. Here's the PR where this was changed: [angular/material.angular.io#641](https://github.com/angular/material.angular.io/pull/641)



trevorhreed commented on Oct 3 • edited ▼

The specific cause of this error for me (in my non-angular project) is the use of `uglifyjs-webpack-plugin`. It uses a ton a memory. For now I disabled minification in production, and will have to figure out how to fix it later.

This is probably related to source maps. Try setting `sourceMap` to `false` in `tsconfig.json` (or, for those with angular projects, in `angular.json`).



BasicExp commented on Oct 9

A Note For Angular/Windows/Docker Users

If you are trying to run 'npm run build' or something of the sort inside your Docker file and you are getting this error, as mention above, you want to use the following run command instead, increase Node.js' access to memory.

```
node --max_old_space_size=8192 node_modules/@angular/cli/bin/ng build --prod
```

In addition, however, Docker is running a Linux VM in the background that will also have memory constraints that need to be addressed.

WARNING

Increasing the VM memory can have a drastic impact on your machine performance, so ensure that you have enough physical memory to do this before trying. If this solution works for you, you should then try to roll back the values as much as you can, by doing a process something like, halving the value initially, testing, halving it again if it works, or adding half of the new value back if it doesn't etc. This will help you minimize the impact on your machine.

You will also want to disable Docker as a start up program for Windows 10. Simply search "startup apps" in the task bar and select the corresponding option that appears. This way, you wont have a VM you are not using bogging down your system.



2

**joelowrance** commented on Oct 9

for me the solution was editing my tsconfig (ionic4 and angular8)

i was playing around adding properties to tsconfig, towards the end i enabled source maps for my prod build inside `angular.json`.

i had to remove the `sourceMap: true` and `sourceBase: "/"` from tsconfig, after that `--prod` works like a charm

Just removing sourceMap worked for me. Frustrating....

**mackelito** referenced this issue on Oct 9**Node version #9**

Closed

**sunilbcone** commented on Oct 14 • edited ▼



mac4281 commented 29 days ago

Had the same issue and upgrading node to 12.12 today let me `ng build --prod`



3



Snesi commented 29 days ago • edited ▼

We're using Bitbucket pipelines and we can't increase the memory limit to 8GB. Also our application takes about 7-10min to build and we need the sourceMaps for our Sentry error reports.

UPDATE: running `ng update @angular/cli @angular/core` solved the problem.



1



TarasKovalenko commented 25 days ago

I updated the node.js to the latest LTS (at this moment 12.13.0) version and everything work perfectly. No need anymore set the `max-old-space-size`.



2



2



BeniFreitag commented 23 days ago

```
"architect": {  
  "build": {  
    "options": {  
      "outputPath": "Y:/apps/dist" // this caused "JavaScript heap out of memory!"  
    }  
  }  
}
```



 magicjj commented 23 days ago • edited ▼

We're using Bitbucket pipelines and we can't increase the memory limit to 8GB. Also our application takes about 7-10min to build and we need the sourceMaps for our Sentry error reports.

UPDATE: running `ng update @angular/cli @angular/core` solved the problem.

Running the `ng update` worked for me as well. One of the minor point versions must have had a memory leak of some sort. Moving from 8.2.8 to 8.2.11 fixed the issue.



1



KingDarBoja referenced this issue 20 days ago

ng-toolkit/universal implementation issue #687


 Closed




martinbrom referenced this issue 8 days ago


fix(datepicker): prevent date input from displaying -1 day for specific timezones #3993


 Merged

 **tailot** commented 7 days ago • edited ▾

With the latest versions of node is solved (at least for me). I recommend the team node to use in the package.json constructs Engines:
<https://docs.npmjs.com/files/package.json#Engines>

 1

 1

 +

Assignees

 filipesilva

Labels

comp: devkit/build-angular
freq3: high
severity4: memory/performance
type: bug/fix

Projects

None yet

Milestone

8.1.x

81 participants



and others