

```
1  command_prompt_1
2  -----
3  D:\test>jdb -XX:+UseSerialGC -Xmx10m -J-Duser.country=en.US
4  Initializing jdb ...
5  > stop in Test.fn
6  Deferring breakpoint Test.fn.
7  It will be set after the class is loaded.
8  > run Main
9  run Main
10 Set uncaught java.lang.Throwable
11 Set deferred uncaught java.lang.Throwable
12 >
13 VM Started: Set deferred breakpoint Test.fn
14
15 Breakpoint hit: "thread=main", Test.fn(), line=5 bci=0
16 5      Test2 t3 = new Test2();
17
18 main[1] next
19
20 Step completed: > "thread=main", Test.fn(), line=6 bci=8
21 6      }
22
23 main[1]
24
25 command_prompt_2
26 -----
27 D:\test>jps
28 4328 Main
29 9064 Jps
30 7716 TTY
31
32 D:\test>java -cp .;%JAVA_HOME%\lib\sa-jdi.jar sun.jvm.hotspot.HSDB
33
34 hsdb_session
35 -----
36 hsdb> universe
37 Heap Parameters:
38 Gen 0: eden [0x0000000fa400000,0x0000000fa4aad68,0x0000000fa6b0000) space capacity = 2818048, 24.831088753633722 used
39 from [0x0000000fa6b0000,0x0000000fa6b0000,0x0000000fa700000) space capacity = 327680, 0.0 used
40 to [0x0000000fa700000,0x0000000fa700000,0x0000000fa750000) space capacity = 327680, 0.0 used
41 Invocations: 0
42
43 Gen 1: old [0x0000000fa750000,0x0000000fa750000,0x0000000fae00000) space capacity = 7012352, 0.0 used
44 Invocations: 0
45
46 perm [0x0000000fae00000,0x0000000fb078898,0x0000000fc2c0000) space capacity = 21757952, 11.90770160721009 used
47 Invocations: 0
48
49 hsdb> scanoops 0x0000000fa400000 0x0000000fc2c0000 Test2
50 0x0000000fa49a710 Test2
51 0x0000000fa49a730 Test2
52 0x0000000fa49a740 Test2
53 Error: sun.jvm.hotspot.debugger.DebuggerException: Windbg Error: ReadVirtual failed!
54 hsdb> revptrs 0x0000000fa49a710
55 Computing reverse pointers...
56 Done.
57 null
58 Oop for java/lang/Class @ 0x0000000fa499b00
59 hsdb> whatis 0x0000000fa499b00
60 Address 0x0000000fa499b00: In thread-local allocation buffer for thread "main" (1)
61 [0x0000000fa48f490,0x0000000fa49a750,0x0000000fa49d118)
62
63 hsdb> inspect 0x0000000fa499b00
64 instance of Oop for java/lang/Class @ 0x0000000fa499b00 @ 0x0000000fa499b00 (size = 120)
65 <<Reverse pointers>>:
66 t1: Oop for Test2 @ 0x0000000fa49a710 Oop for Test2 @ 0x0000000fa49a710
67 hsdb> revptrs 0x0000000fa49a730
68 Oop for Test @ 0x0000000fa49a720
69 hsdb> whatis 0x0000000fa49a720
70 Address 0x0000000fa49a720: In thread-local allocation buffer for thread "main" (1)
71 [0x0000000fa48f490,0x0000000fa49a750,0x0000000fa49d118)
72
73 hsdb> inspect 0x0000000fa49a720
74 instance of Oop for Test @ 0x0000000fa49a720 @ 0x0000000fa49a720 (size = 16)
75 <<Reverse pointers>>:
76 _mark: 1
77 t2: Oop for Test2 @ 0x0000000fa49a730 Oop for Test2 @ 0x0000000fa49a730
78 hsdb> revptrs 0x0000000fa49a740
79 null
80 hsdb> mem 0x0000000287f7f0 30
81 0x0000000287f7f0: 0x00000002886298
82 0x0000000287f7f8: 0x00000002893ca5
83 0x0000000287f800: 0x00000002893ca5
84 0x0000000287f808: 0x0000000287f808
85 0x0000000287f810: 0x0000000fb077f58
86 0x0000000287f818: 0x0000000287f860
87 0x0000000287f820: 0x0000000fb078360
88 0x0000000287f828: 0x0000000000000000
89 0x0000000287f830: 0x0000000fb077f78
90 0x0000000287f838: 0x0000000000000000
```

```
85 0x00000000287f840: 0x00000000287f860
86 0x00000000287f848: 0x00000000287f8a8
87 0x00000000287f850: 0x000000002886298
88 0x00000000287f858: 0x00000000fa49a740
89 0x00000000287f860: 0x00000000fa49a720
90 0x00000000287f868: 0x00000000287f868
91 0x00000000287f870: 0x00000000fb077039
92 0x00000000287f878: 0x00000000287f8c0
93 0x00000000287f880: 0x00000000fb077350
94 0x00000000287f888: 0x0000000000000000
95 0x00000000287f890: 0x00000000fb077060
96 0x00000000287f898: 0x00000000287f860
97 0x00000000287f8a0: 0x00000000287f8c0
98 0x00000000287f8a8: 0x00000000287f9a0
99 0x00000000287f8b0: 0x00000000288062a
100 0x00000000287f8b8: 0x00000000fa49a720
101 0x00000000287f8c0: 0x00000000fa498ea8
102 0x00000000287f8c8: 0x0000000000000000
103 0x00000000287f8d0: 0x0000000000000000
104 0x00000000287f8d8: 0x0000000000000000
105
106 hsd> jstack -v
107 Deadlock Detection:
108
109 No deadlocks found.
110
111 Thread 6: (state = BLOCKED, current Java SP = null
112 )
113
114 Locked ownable synchronizers:
115 - None
116
117 Thread 5: (state = BLOCKED, current Java SP = null
118 )
119
120 Locked ownable synchronizers:
121 - None
122
123 Thread 4: (state = BLOCKED, current Java SP = 0x000000005f6ef50
124 )
125 - java.lang.Object.wait(long) @bci=0, pc=0x000000002889c61, methodOop=0x00000000fae02958 (Interpreted frame)
126 - java.lang.ref.ReferenceQueue.remove(long) @bci=44, line=135, pc=0x000000002886298, methodOop=0x00000000faed0a60 (Interpreted frame)
127 - java.lang.ref.ReferenceQueue.remove() @bci=2, line=151, pc=0x000000002886473, methodOop=0x00000000faed0b40 (Interpreted frame)
128 - java.lang.ref.Finalizer$FinalizerThread.run() @bci=3, line=177, pc=0x000000002886473, methodOop=0x00000000faed21d8 (Interpreted frame)
129
130 Locked ownable synchronizers:
131 - None
132
133 Thread 3: (state = BLOCKED, current Java SP = 0x000000005a1f730
134 )
135 - java.lang.Object.wait(long) @bci=0, pc=0x000000002889c61, methodOop=0x00000000fae02958 (Interpreted frame)
136 - java.lang.Object.wait() @bci=2, line=503, pc=0x000000002886298, methodOop=0x00000000fae02b68 (Interpreted frame)
137 - java.lang.ref.Reference$ReferenceHandler.run() @bci=46, line=133, pc=0x000000002886298, methodOop=0x00000000faecf620 (Interpreted frame)
138
139 Locked ownable synchronizers:
140 - None
141
142 Thread 1: (state = BLOCKED, current Java SP = 0x00000000287f808
143 )
144 - Test.fn() @bci=8, line=6, pc=0x000000002893ca5, methodOop=0x00000000fb077f78 (Interpreted frame)
145 - Main.main(java.lang.String[]) @bci=9, line=4, pc=0x000000002886298, methodOop=0x00000000fb077060 (Interpreted frame)
146
147 Locked ownable synchronizers:
148 - None
149
150 hsd> inspect 0x00000000fb077f58
151 Error: sun.jvm.hotspot.debugger.UnalignedAddressException: 100011
152 hsd> inspect 0x00000000fb077f78
153 instance of Method fn()V@0x00000000fb077f78 @ 0x00000000fb077f78 @ 0x00000000fb077f78 (size = 136)
154 _mark: 1
155 _constMethod: ConstMethod fn()V@0x00000000fb077f08 @ 0x00000000fb077f08 Oop @ 0x00000000fb077f08
156 _constants: ConstantPool for Test @ 0x00000000fb077c68 Oop @ 0x00000000fb077c68
157 _method_size: 17
158 _max_stack: 2
159 _max_locals: 2
160 _size_of_parameters: 1
161 _access_flags: 1
162 hsd> inspect 0x00000000fb077f08
163 instance of ConstMethod fn()V@0x00000000fb077f08 @ 0x00000000fb077f08 @ 0x00000000fb077f08 (size = 112)
164 _mark: 1
165 _method: Method fn()V@0x00000000fb077f78 @ 0x00000000fb077f78 Oop @ 0x00000000fb077f78
166 _exception_table: [I @ 0x00000000fae01d50 Oop for [I @ 0x00000000fae01d50
167 _constMethod_size: 14
168 _flags: 5
169 _code_size: 9
```

```

170     _name_index: 18
171     _signature_index: 12
172     _generic_signature_index: 0
173     _code_size: 9
174     hsdb> whatis 0x0000000fb077f58
175     Address 0x0000000fb077f58: In perm generation    perm [0x0000000fae00000,0x0000000fb078898,0x0000000fc2c0000) space capacity = 21757952,
11.90770160721009 used
176     hsdb> print 0x0000000fb077f78
177
178     public void fn() @0x0000000fb077f78
179
180     Holder Class
181     public class Test @0x0000000fb078100
182     Bytecode
183     line bci    bytecode
184     5    0    new #2 [Class Test2]
185     5    3    dup
186     5    4    invokespecial #1 [Method Test2.<init>()V]
187     5    7    astore_1
188     6    8    return
189
190
191     Constant Pool
192     Constant Pool of [public class Test @0x0000000fb078100] @0x0000000fb077c68
193
194     hsdb> mem 0x0000000fb077f50 2
195     0x0000000fb077f50: 0x4c0001b7590200ca
196     0x0000000fb077f58: 0x0000000004105b1
197
198     hsdb> class Test
199     Test @0x0000000fb078100
200     hsdb> print 0x0000000fb078100
201
202     public class Test @0x0000000fb078100
203
204
205
206     Super Class
207     public class java.lang.Object @0x0000000fae02df8
208     Fields
209
210     static    Test2 t1; (offset = 112)
211     Test2 t2; (offset = 12)
212
213
214     Methods
215     public void <init>() @0x0000000fb077e80;
216     static void <clinit>() @0x0000000fb078058;
217     public void fn() @0x0000000fb077f78;
218
219     Constant Pool
220     Constant Pool of [public class Test @0x0000000fb078100] @0x0000000fb077c68
221     hsdb>
222
223     Main.java
224     -----
225     public class Main {
226     public static void main(String[] args) {
227         Test test = new Test();
228         test.fn();
229     }
230 }
231
232     Main_main_bytecode
233     -----
234     hsdb> mem 0x0000000fb077030 2
235     0x0000000fb077030: 0x4c0001b7590200bb
236     0x0000000fb077038: 0x214103b10002b62b
237
238     0x0000000fb077030: bb 00 02    new <cp index #2> [Class Test]
239     0x0000000fb077033: 59          dup
240     0x0000000fb077034: b7 01 00    invokespecial <cp cache index #1> [Method Test.<init>()V]
241     0x0000000fb077037: 4c          astore_1
242     0x0000000fb077038: 2b          aload_1
243     0x0000000fb077039: b6 02 00    invokevirtual <cp cache index #2> [Method Test.fn()V]
244     0x0000000fb07703c: b1          return
245
246     stack_frame_layout
247     -----
248     0x00000000287f7f0: 0x000000002886298
249     0x00000000287f7f8: 0x000000002893ca5
250     0x00000000287f800: 0x000000002893ca5
251     -----
252     Stack frame for Test.fn() @bci=8, line=6, pc=0x000000002893ca5, methodOop=0x0000000fb077f78 (Interpreted frame)
253     0x00000000287f808: 0x00000000287f808 expression stack bottom    <- rsp
254     0x00000000287f810: 0x00000000fb077f58 bytecode pointer          = 0x0000000fb077f50 (base) + 8 (bytecode index) in PermGen

```

```
255 0x00000000287f818: 0x00000000287f860 pointer to locals
256 0x00000000287f820: 0x00000000fb078360 constant pool cache = ConstantPoolCache for Test in PermGen
257 0x00000000287f828: 0x0000000000000000 method data oop = null
258 0x00000000287f830: 0x00000000fb077f78 method oop = Method for Test.fn()V in PermGen
259 0x00000000287f838: 0x0000000000000000 last Java stack pointer (not set)
260 0x00000000287f840: 0x00000000287f860 old stack pointer (saved rsp)
261 0x00000000287f848: 0x00000000287f8a8 old frame pointer (saved rbp) <- rbp
262 0x00000000287f850: 0x000000002886298 return address = in interpreter codelet "return entry points" [0x0000000028858b8,
0x0000000028876c0] 7688 bytes
263 0x00000000287f858: 0x00000000fa49a740 local[1] "t3" = Oop for Test2 in NewGen
264 0x00000000287f860: 0x00000000fa49a720 local[0] "this" = Oop for Test in NewGen
-----
266 0x00000000287f868: 0x00000000287f868
267 0x00000000287f870: 0x00000000fb077039
268 0x00000000287f878: 0x00000000287f8c0
269 0x00000000287f880: 0x00000000fb077350
270 0x00000000287f888: 0x0000000000000000
271 0x00000000287f890: 0x00000000fb077060
272 0x00000000287f898: 0x00000000287f860
273 0x00000000287f8a0: 0x00000000287f8c0
274 0x00000000287f8a8: 0x00000000287f9a0
275 0x00000000287f8b0: 0x00000000288062a
276 0x00000000287f8b8: 0x00000000fa49a720
277 0x00000000287f8c0: 0x00000000fa498ea8
278 0x00000000287f8c8: 0x0000000000000000
279 0x00000000287f8d0: 0x0000000000000000
280 0x00000000287f8d8: 0x0000000000000000
281
282
283 0x00000000287f7f0: 0x000000002886298
284 0x00000000287f7f8: 0x000000002893ca5
285 0x00000000287f800: 0x000000002893ca5
286 0x00000000287f808: 0x00000000287f808
287 0x00000000287f810: 0x00000000fb077f58
288 0x00000000287f818: 0x00000000287f860
289 0x00000000287f820: 0x00000000fb078360
290 0x00000000287f828: 0x0000000000000000
291 0x00000000287f830: 0x00000000fb077f78
292 0x00000000287f838: 0x0000000000000000
293 0x00000000287f840: 0x00000000287f860
294 0x00000000287f848: 0x00000000287f8a8
295 0x00000000287f850: 0x000000002886298
296 0x00000000287f858: 0x00000000fa49a740
-----
298 Stack frame for Main.main(java.lang.String[]) @bci=9, line=4, pc=0x000000002886298, methodOop=0x00000000fb077060 (Interpreted frame)
299 0x00000000287f860: 0x00000000fa49a720 expression stack[0] = Oop for Test in NewGen
300 0x00000000287f868: 0x00000000287f868 expression stack bottom
301 0x00000000287f870: 0x00000000fb077039 bytecode pointer = 0x00000000fb077030 (base) + 9 (bytecode index) in PermGen
302 0x00000000287f878: 0x00000000287f8c0 pointer to locals
303 0x00000000287f880: 0x00000000fb077350 constant pool cache = ConstantPoolCache for Main in PermGen
304 0x00000000287f888: 0x0000000000000000 method data oop = null
305 0x00000000287f890: 0x00000000fb077060 method oop = Method for Main.main([Ljava/lang/String;)V in PermGen
306 0x00000000287f898: 0x00000000287f860 last Java stack pointer
307 0x00000000287f8a0: 0x00000000287f8c0 old stack pointer
308 0x00000000287f8a8: 0x00000000287f9a0 old frame pointer
309 0x00000000287f8b0: 0x00000000288062a return address = in StubRoutines
310 0x00000000287f8b8: 0x00000000fa49a720 local[1] "test" = Oop for Test in NewGen
311 0x00000000287f8c0: 0x00000000fa498ea8 local[0] "args" = Oop for java.lang.String[] in NewGen
-----
313 0x00000000287f8c8: 0x0000000000000000
314 0x00000000287f8d0: 0x0000000000000000
315 0x00000000287f8d8: 0x0000000000000000
316
317 Test.java
318 -----
319 hsdb> mem 0x00000000fb077f50 2
320 0x00000000fb077f50: 0x4c0001b7590200ca
321 0x00000000fb077f58: 0x0000000004105b1
322
323 0x00000000fb077f50: bb 00 02 new <cp index #2> [Class Test2]
324 0x00000000fb077f53: 59 dup
325 0x00000000fb077f54: b7 01 00 invokespecial <cp cache index #1> [Method Test2.<init>()V]
326 0x00000000fb077f57: 4c astore_1
327 0x00000000fb077f58: b1 return
328
```