

# How to report IntelliJ IDEA performance problems and take CPU snapshots

---

## Common solutions for IntelliJ IDEA performance problems

Before capturing and providing the CPU snapshot, please be aware of the following known issues (and possible solutions):

- 
- **Windows System Restore** locking attrib.dat.data IntelliJ IDEA cache file when **idea.system.path** is set to custom location. Disable System Restore. (This should be obsolete starting from IntelliJ IDEA **7.0.4** as now IntelliJ IDEA uses different extension for such files and they are no longer affected by the System Restore).
- **Antiviral tools** monitoring project and caches directories (USER\_HOME\IntelliJdeaXX). Exclude IntelliJ IDEA project/library/caches directories from the monitor. In some cases only complete uninstall will fix the problem because of the low level drivers affecting the performance even when you disable the antivirus.
- IntelliJ IDEA caches on the **slow or network drive**, check that USER\_HOME\IntelliJdeaXX is on the local drive, adjust the location in IDEA\_HOME\bin\idea.properties if necessary (but be aware of the first item in this list).
- Project files/JDK files/Library jar files on the **network drives**. Move them to your local drives.
- **Low free system memory** and therefore swapping. Close other applications, free as much memory as possible. Install additional RAM, decrease the heap.
- Very large files (>10 000 lines). Refactor them if possible. Disable inspections per file to save some resources using the Hector icon in the status bar.
- Heavily **fragmented partition** and full MFT, see the [related blog post](#). Drive defragmentation can make IntelliJ IDEA much faster, we have proven it on our development machines.
- **Heavy disk activity** caused by another applications, such as torrent clients, file servers. One thread can read data from the modern HDD at ~100 MB/sec. If there are 2 or more applications reading/writing to the same HDD at the same time, performance will drop to 2-3 MB/sec. Check that there are no processes constantly reading/writing to the disk. On Windows you can use SysInternals [Process Monitor](#) or [Process Explorer](#).
- Ant/Maven build or other **external process modifying or generating lots of files/logs into the project directories**. Synchronization will take lots of time in this case. Exclude such directories from the project.
- When using version control, if there are many **unversioned files** (thousands), Changes view update may be slow. Exclude such directories from the version control in Settings | Version Control | Directory Version Control Settings by adding the directory with such files to the list and setting VCS to NONE.
- **Too many nodes open in the Project View** will slow down editing. Collapsing the nodes will improve the performance. This issue is addressed in IntelliJ IDEA 9.0.
- On **Linux** the editor performance may be seriously affected by the **video adapter/drivers**. It can be improved by running with `-Dsun.java2d.pmoscreeen=false` VM option or by [editing the xorg.conf](#).
- **Too low heap size** when running under **64-bit JVM** or when working with **large projects**. Low heap will cause frequent garbage collection. Increase the heap in `idea.exe.vmoptions` (Windows), `idea.vmoptions` (Linux) or `Info.plist` (Mac). For 64-bit JVM you may need to double the defaults. Adding `-XX:+UseCompressedOops` option should also help when running under 64-bit JDK. Note that normally you should not use heap size higher than 1GB, especially on the 32-bit JVM, where the VM may fail to start with the heap `(-Xmx) > 1GB`. If the heap is too high so that it doesn't fit into the free physical RAM, your OS will start swapping which will reduce the performance, also GC pauses may become longer. Don't use extreme memory values unless you know what you are doing.

## Automatic Thread Dumps Logging

If the UI of IntelliJ IDEA is unresponsive for more than 5 seconds, it starts writing thread dumps to the logs directory. If you've encountered a situation of IntelliJ IDEA unresponsiveness or deadlock, please find the **threadDumps-xxx** directory from the session in which you've experienced the problem and send the contents to [support@jetbrains.com](mailto:support@jetbrains.com).

The logs directory on Windows and Linux is *USERPROFILE\IntelliJ\ideaXX\system\log*.

On Mac OS X, the logs directory can be found at:

~/Library/Caches/IntelliJIDEAXX/logs

for IntelliJ IDEA 8.x and older versions

~/Library/Logs/IntelliJIDEAXX

for 9.0 and later IntelliJ IDEA versions

## CPU snapshot capturing instructions

If you have a performance problem not related to the issues mentioned above, please contact [support@jetbrains.com](mailto:support@jetbrains.com) and provide the following information:

- You hardware configuration (CPU, Memory, HDD)
- OS, free memory when you start IntelliJ IDEA, IntelliJ IDEA heap settings, file system, IntelliJ IDEA caches location (if modified)

**IntelliJ IDEA Ultimate** includes a built-in YourKit profiling agent that can be used to take CPU and memory snapshots. Please use the following instructions to enable it.

Open **IDEA\_HOME/bin/idea.exe.vmoptions** (idea.vmoptions on Linux/Solairs), add the following line at the bottom:

```
-agentlib:yjpagent
```

On **Mac OS X** open **Info.plist** located in **/Applications/IntelliJ IDEA X.X.app/Contents**, find the following:

```
<key>VMOptions</key>
```

```
<string>-Xms16m -Xmx192m -XX:MaxPermSize=120m -Xbootclasspath/p:../lib/boot.jar -ea</string>
```

Modify the string tag by adding **-agentlib:yjpagent** option, like:

```
<string>-Xms16m -Xmx192m -XX:MaxPermSize=120m -Xbootclasspath/p:../lib/boot.jar -ea -agentlib:yjpagent</string>
```

If you want to **profile the slow startup** of the application, CPU snapshot recording should be started automatically by using the following option instead:

```
-agentlib:yjpagent=sampling,noj2ee
```

Note that **yjpagent** library provided with IntelliJ IDEA 9 or earlier **will not work on 64-bit operating system** when you run IntelliJ IDEA **under 64-bit JVM**. If you can't run under 32-bit JVM, please obtain the 64-bit library version from the [YourKit Java Profiler](#) distribution (you need Linux **.zip** download of **7.5 or 8.x version** containing library versions for different operating systems: Linux, Mac, Solaris, Windows). Replace **yjpagent** library in **IDEA\_HOME**

**bin** with the appropriate version. Use **8.x YourKit version** with **IntelliJ IDEA 9.x** and **7.5** version with previous IntelliJ IDEA versions. **IntelliJ IDEA 10 and later** has both 32-bit and 64-bit libraries bundled, therefore this step is not required.

Start IntelliJ IDEA and you should notice 2 new buttons in the toolbar, one for taking **CPU snapshot**, another one for Memory snapshot. Press the CPU button and perform the actions which take a lot of CPU resources on your machine. When finished, press this button again to stop recording, save the snapshot and upload it to <ftp://ftp.intellij.net/uploads/> (please note that listing is not allowed in this directory, only upload is permitted, therefore you will not be able to see/download your and other files there, some FTP clients may not like it, also be sure to switch into the binary transfer mode if uploading via command line ftp client). If you can't access FTP, use some free file sharing service and send us a link.

Send us an e-mail with the file name and we'll investigate the issue. In the e-mail please also describe what you were doing in detail. If you can reproduce it in a small project, please provide one. You can also submit an issue to [YouTrack](#) instead of contacting us by mail.

**IntelliJ IDEA Community Edition** does not include a YourKit profiling agent, because the YourKit agent library is not open-source. To take a CPU or memory snapshot with IntelliJ IDEA Community Edition, you can download an evaluation version of [YourKit Java Profiler](#), copy the yjagent library to the IntelliJ IDEA binary directory, add it to the VM options as described above, and connect to the running IntelliJ IDEA instance from the standalone YourKit Profiler UI.

## Memory snapshot

In case of memory related issues (memory usage goes high, garbage is not collected, etc) please use the Memory snapshot button in the menu near the CPU snapshot button. If it's not possible to get the snapshot because of the application crashing with OutOfMemory errors, please add the

**-XX:+HeapDumpOnOutOfMemoryError**

option to the IntelliJ IDEA JVM options. On the next OOM error the .hrpof dump will be produced and saved by the JVM (usually in the application working directory which is IDEA\_HOME\bin). Upload this dump to our FTP as described above in the CPU snapshot section.

Please note that memory snapshot may contain the sensitive source code from your project. If you are uploading to a public service, use some password protection or encryption. JetBrains FTP server is write only and you don't need to protect files uploaded there.