

Object Instance Tracking for Java


On this page:

- [Prerequisites for Object Instance Tracking](#)
- [Specifying the Classpath](#)
- [Starting Object Instance Tracking](#)
- [Tracking Specific Classes](#)

This topic helps you understand how to configure and use object instance tracking. For more information about why you may need to configure this, see [Java Memory Thrash](#).

When object instance tracking is enabled, AppDynamics tracks the top 20 application and top 20 system (core Java) classes in the heap by default. You can configure specific classes to track as described below.

Enabling object instance tracking increases the amount of information captured by the agents, resulting in additional overhead. AppDynamics recommends using object instance tracking only while troubleshooting potential memory leaks. It does not normally need to be enabled during normal operation.

 AppDynamics does not perform allocation tracking for core Java classes by default, since doing so would add significant system overhead. However, it is possible to track core classes on a short-term basis (for example, while troubleshooting) or in pre-production environments. For more information, see the knowledge base article [Enabling Allocation Tracking for Core Java Classes](#).

Permissions

To enable object instance tracking, you need the Configure Agent Properties permission.

To configure the custom classes to track, you need the Configure Memory Monitoring permission.

For information on AppDynamics Role Based Access Control, see [Roles and Permissions](#).

Prerequisites for Object Instance Tracking

- Please check [Java Supported Environments](#) for information about platform support for object instance tracking.
- Object instance tracking uses tools.jar. If your application runs with the JDK, tools.jar should be already available. If you are running with the JRE, you must add tools.jar to JRE_HOME/lib/ext and restart the JVM. You can find tools.jar in JAVA_HOME/lib/tools.jar.
- In some cases you might also need to copy libattach.so (Linux) or attach.dll (Windows) from your JDK to your JRE.
- Depending on the JDK version, you may also need to specify the classpath as shown below (along with other -jar options).

Specifying the Classpath

When using the JDK runtime environment, set the classpath using the -classpath option for the application. For example:

- On Windows:
`java -classpath <complete-path-to-tools.jar>;%CLASSPATH% -jar myApp.jar`
- On Unix:
`java -Xbootclasspath/a:<complete-path-to-tools.jar> -jar myApp.jar`

Starting Object Instance Tracking

To start an object instance tracking session, follow these steps:

1. In the left navigation pane, click **Tiers & Nodes**.
2. In the right pane, expand the tier node and open the Node Dashboard for the node on which you want to enable object instance tracking.

3. Click the Memory tab.
4. Click the **Object Instance Tracking** subtab.
5. Click **ON**.

Tracked classes now appear in the Object Instance Tracking table. You can drill down to the tracked classes to see details.

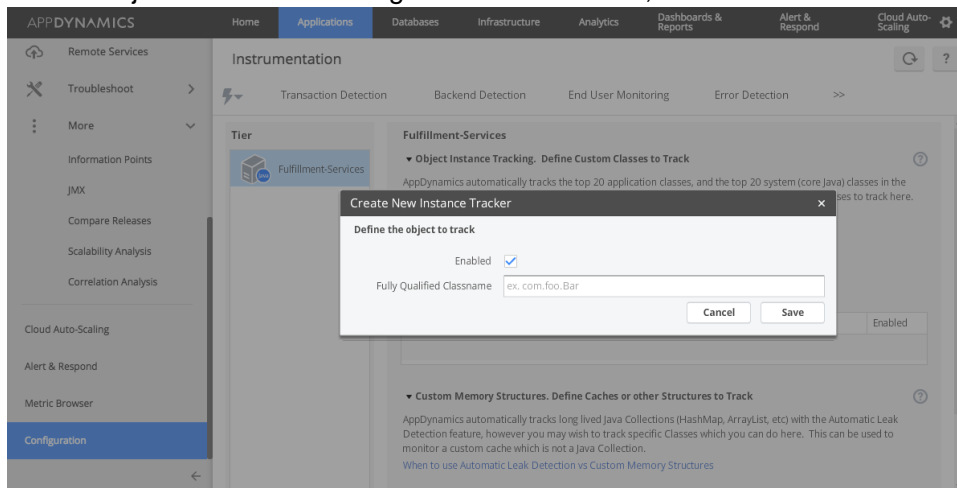
Tracking Specific Classes

For performance reasons, only the top 20 application classes and the top 20 system (core Java) classes in the heap are tracked automatically.

Use the "Configure Custom Classes to Track" option to specify instances of specific classes to track. Note that the classes you configure here are tracked only if their instance count is among the top 1000 instance counts in the JVM.

To track instances of custom classes

1. Click **Configuration > Instrumentation > Memory Monitoring**.
2. In the "Object Instance Tracking" section for the tier, click **Add**.



3. Enter the fully-qualified class name of the class to track and click **Save**.

The class you added is now tracked during object instance tracking sessions.