Performance (Copy)

When you report an issue with your modeling tool, Cameo Collaborator or Teamwork Cloud, always include related log files. Log files provide useful information that we need to investigate and troubleshoot your problem.

Accessing log files

- **For a modeling tool:**

Open your modeling tool and in the main menu go to **Help** > **About <modeling tool name>**. In the **Environment** tab, click the link provided in the **Log File** line to open a log file.

Or

Go to *<user.home>\<.modeling tool name>\<modeling tool version>* directory and obtain *<modeling tool name>.log* file.

Default path for Windows*: C:\Users\<USERNAME>\AppData\Local\.<modeling tool name>\<modeling tool version number>*
Default path for other operating systems: *<user.home>/.<modeling tool name>/< modeling tool version number>*

- **For WebApp/Cameo Collaborator for Teamwork Cloud:**

Go to *<WebAppPlatform installation directory>/logs/webappplatform* directory, and obtain the *web-app.log* and/or *collaborator.log* files.

Default path for Linux: */opt/local/TeamworkCloud/WebAppPlatform/logs/webappplatform*

Default path for Windows: *C:\Program Files\TeamworkCloud\WebAppPlatform\logs\webappplatform*

- **For Teamwork Cloud:**

Go to the *<user.home>\.twcloud\<version number>* directory and obtain the *server.log* file.

Default path for Linux: */home/twcloud/.twcloud/<version>*
Default path for Windows: *C:\Users\<USER>\.twcloud\<version>*

Go to the *<user.home>\.authserver\<version number>* directory and obtain the *authserver.log* file.

Default path for Linux: */home/twcloud/.authserver/<version>*
Default path for Windows: *C:\Users\<USER>\.authserver\<version>*

- **For Cassandra:**

Go to *<Cassandra installation directory>\logs* directory and obtain the *cassandra.log* file.

Default path for Linux: */var/log/cassandra*
Default path for Windows: *C:\apache-cassandra-<version>\logs*

- **Properties files for Teamwork Cloud:**

Default for Linux:
/opt/local/TeamworkCloud/configuration/application.conf
/opt/local/TeamworkCloud/Authserver/config/authserver.properties
/opt/local/TeamworkCloud/WebAppPlatform/shared/conf/webappplatform.properties

Default for Windows:
C:\Program Files\TeamworkCloud\configuration\application.conf
C:\Program Files\TeamworkCloud\Authserver\config\authserver.properties
C:\Program Files\TeamworkCloud\WebAppPlatform\shared\conf\webappplatform.properties

- **For Cameo Collaborator for Alfresco:**

Go to Alfresco Community installation directory, and obtain the *alfresco.log*, *share.log* and *solr.log* files.

Go to *<Alfresco Community installation directory>\tomcat\logs*, and obtain the *alfrescotomcat-stderr.log, alfrescotomcat-stdout.log* and *catalina.log* files.

Generating a log file

If the performance of your modeling tool declines or it freezes, go to *<modeling tool installation directory>\bin,* and run the *submit_issue.exe* or *submit_issue.sh* file several times. It dumps threads into the log file of your modeling tool. Then you can access the log file as described above.

Java crash log files

If your modeling tool crashes (disappears), try searching for Java crash log files. They are stored in the running location of the modeling tool, e.g., the *<modeling tool installation directory>/bin* or *<modeling tool installation directory>* directory. The names of these log files start with "hs_err", e.g., *hs_err_pid15693.log*. If you find the Java crash log file, it indicates that Java was the reason of your modeling tool crash.
In order to troubleshoot this issue, we will need the program's log files (located in the install directory of your tool). Additional information might be required about the Java environment on which our modeling tools are based. The information we need is called the "heapdump" and can be obtained using the Java VisualVM program.

Please read and follow this procedure to install Java VisualVM and send this data to us (while the tool is running):

1. Download and install the **Java VisualVM** with **Java SE Development Kit (JDK**, valid till version 1.8**)** or separately from https://visualvm.github.io/, if you do not have it.
2. Start Task Manager or other appropriate tool depending on your OS.
3. Start Java VisualVM. If you have selected the default location on the JDK installation process, VisualVm is located in C:\Program Files\Java\jdk<version number>\bin\ jvisualvm.exe. Otherwise start the jvisualvm.exe from your custom location.
4. In the Task Manager or other program according to you OS, find the PID (Process Identifier) of your modeling tool.
5. In the Java VisualVM, find Java process by the modeling tool PID and double click to open it.
6. Right click the Java process and, from the shortcut menu, select **Heap Dump**. The heapdump file creates under your process.
7. Save the heapdump file. Right click the heapdump and, on the shortcut menu, click **Save As**.
8. Register an issue and sent the saved *.hprof file for the investigation. For this, on the main menu of your modeling tool, click **Help** > **Report an Issue**

To solve the problem, we need a little input from you. We need to examine the log files to troubleshoot the problem. The following list outlines how to submit log files:

Generating a log file when the modeling tool freezes or performs slowly:

1. When a modeling tool runs slowly or freezes, open *<modeling tool installation directory>\bin,* and run *submit_issue.exe* several times. It dumps threads into the log file.

2. Click **Help** > **About** on the main menu of your modeling tool. In the open dialog, select the **Environment** tab and click the **Log File** link. The log file opens.*

3. Save the file and register an issue. Click **Help** > **Report an Issue** on the main menu of your modeling tool

*If a modeling tool is inactive, you can find the log file under:

- For Windows OS: C:\Users\<USERNAME>\AppData\Local\.<modeling tool name>\<modeling tool version number>
- For Other OS: <user.home>/.<modeling tool name>/< modeling tool version number>

By using Java VisualVM:

Our modeling tools are Java-based; thus, you can use the Java VisualVM program for performance issue examination. Data provided by VisualVM may help to explore issues accurately.

The following steps outline how to obtain the data from VisualVM:

Note. Please read the steps first to familiarize yourself with the whole procedure to gather more precise information. Step #8 should be started as soon as possible.

On Windows OS:

1. Download and install the **Java VisualVM** with **Java SE Development Kit (JDK**, valid till version 1.8**)** or separately from https://visualvm.github.io/.
2. Start **Task Manager**.
3. Start **Java VisualVM**. If you selected the default location on the JDK installation process, VisualVM is located in *C:\Program Files\Java\jdk<version number>\bin\ jvisualvm.exe*. Otherwise, start the *exe* from your customized location.
4. Start the modeling tool.
5. In **Task Manager**, find the PID (Process Identifier) of your modeling tool.
6. In **Java VisualVM**, find the Java process by the modeling tool PID in the Applications tree on the left (the same PID as in **Task Manager**) and double-click to open it.
7. Click the **Sampler** tab and click the **CPU**.
8. Initiate the action causing the low performance of your modeling tool.
9. Wait until that action in your modeling tool is finished, then click the **Stop button**.
10. In the **CPU samples** tab, click the **Snapshot button**. The snapshot is created in the **Applications** tree on the left.
11. To save the snapshot, right-click it and select **Save As to save the *.*nps* file**.
12. Raise a support ticket via 3DSupport application and attach the .nps file to the ticket.

1. Download and install the **Java VisualVM** with **Java SE Development Kit (JDK**, valid till version 1.8**)** or separately from https://visualvm.github.io/.
2. Start **Activity Monitor**.
3. Start **Java VisualVM**. If you selected the default location on the JDk installation process, VisualVM is located in */Library/Java/JavaVirtualMachines /jdk<version number>.jdk /Contents/Home/bin/jvisualvm*. Otherwise, start ***jvisualvm*** from your customized location.
4. Start the modeling tool.
5. In **Activity Monitor**, find the PID (Process Identifier) of your modeling tool.
6. In **Java VisualVM**, find the Java process by the modeling tool PID in the Applications tree on the left (the same PID as in **Activity Monitor**) and double-click to open it.
7. Click the **Sampler** tab and click the **CPU**.
8. Initiate the action causing the low performance of your modeling tool.
9. Wait until that action in your modeling tool is finished and click the Stop button.
10. In the CPU **samples** tab, click the Snapshot button. The snapshot is created in the Applications tree on the left.
11. To save the snapshot, right-click it and select Save As to save the *.*nps* file.
12. Raise a support ticket via 3DSupport application and attach the .nps file to the ticket.

On Linux OS:

1. Download and install the **Java VisualVM** with **Java SE Development Kit (JDK**, valid till version 1.8**)** or separately from https://visualvm.github.io/.
2. Start **System Monitor**.
3. To start **Java VisualVM**, execute the **jvisualvm** tool from the **bin** directory of the JDK. When the tool runs, the **Java VisualVM** window opens.
4. Start the modeling tool.

5. In **System Monitor**, find the PID (Process Identifier) of your modeling tool.
6. In **Java VisualVM**, find the Java process by the modeling tool PID in the Applications tree on the left (the same PID as in **System Monitor**) and double-click to open it.
7. Click the **Sampler** tab and click the **CPU**.
8. Initiate the action causing the low performance of your modeling tool.
9. Wait until that action in your modeling tool is finished, then click the **Stop button**.
10. In the **CPU samples** tab, click the **Snapshot button**. The snapshot is created in the **Applications** tree on the left.
11. To save the snapshot, right-click it and select **Save As to save the *.*nps* file**.
12. Raise a support ticket via 3DSupport application and attach the .nps file to the ticket.

While performing small ordinary tasks (not individual large operations, such as opening a large project or merging several large projects), modeling tool memory leaks may cause OutOfMemory exception errors. This is due to the cumulative effect of performing many small operations that normally do not cause any problems.

In order to analyze and fix the issue, more information is required. Please provide us with the following information:

1. Can you reproduce the problem? If yes, what are the steps to reproduce it?
2. If it is possible to reproduce it, send us the project file. We guarantee confidentiality, and shall sign a Non-disclosure Agreement at your request.
3. Send us the memory dump file. It may help to detect the problem.

To create a memory dump file, please do the following:

1. In the properties* file, append JAVA_ARGS parameters as follows:
   -XX\:+HeapDumpOnOutOfMemoryError -XX\:HeapDumpPath\=d\:\\snapshots
2. Start your modeling tool.

The dump file will be created in the provided location – d:\\snapshots (please make sure that this location is available), if an OutOfMemory exception occurs. As the file may be large, please zip the created file, and upload it to the provided ftp server.

---

*

- For MagicDraw, the properties file is *magcdraw.properties* (*mduml.properties* for MagicDraw version 17.0.4 or earlier).
- For Cameo Enterprise Architecture, the properties files is *cameoea.properties*.
- For Cameo Systems Modeleer, the properties files is *csm.properties*.
- For Cameo Business Modeler, the properties file is *csm.properties*.

This issue appears when MagicDraw Look and Feel is Windows. Please change the Look and Feel to Metal. For this, select **Options** > **Look and Feel** > **M etal**.
If you are running the Sun's JVM, set the following java properties in your mduml.properties* file: in the line JAVA_ARGS=-Xmx600M, change the number of heap size '600' to maximal heap size in megabytes.

For example:

JAVA_ARGS=-Xmx800M

This sets java heap size to 800 megabytes.

If you get exception "java.lang.OutOfMemoryError: PermGen space" please, check if PermSize is specified in mduml.properties* file.

The JAVA_ARGS line should look like:

JAVA_ARGS=-Xmx600M -XX\:PermSize\=40M -XX\:MaxPermSize\=150M

If PermSize is specified in mduml.properties* file and the same problem still appears, the MaxPermSize should be increased. PermSize is part of heap size, so MaxPermSize should always be smaller than heap size specified with Xmx parameter.

For example:

JAVA_ARGS=-Xmx600M -XX\:PermSize\=40M -XX\:MaxPermSize\=200M

* If you are using MagicDraw 17.0.5 or later, the name of the property file is magicdraw.properties.
In cases when MagicDraw is not responding, please, run submit_issue.exe. You may find it in /bin/ folder. In case there are processes, submit issue produces thread dump for it and writes it to md.log file. You can grab thread dump directly from Report an Issue frame, "MD log file" tab. Please attach thread dump to this issue report or send us md.log file. This log can be found in the user home directory (/.magicdraw/).
We noticed such behavior under Windows NT when mapped network drives are present, but a portion of them are offline. Try disconnecting all offline drives.
When you work with very large models or use a lot of diagrams at a time, the performance of MagicDraw may become slow. To increase an efficiency of modeling, we suggest the following solutions:

- Increase a java heap size.
- Do not keep unused diagrams open. Open project without loading diagrams. Your projects will be opened over a shorter period of time without opening a diagram as well as use less memory.
- Increase an active validation period. MagicDraw takes less memory with increased active validations period.
- Split the project to read only modules. Keep read only modules not loaded. This may help only if your project contains several parts with minimal dependencies between them.

- Use Garbage Collector to free unused memory.
- Turn of antivirus. Some antivirus software can cause significant performance decrease on project open and other actions.

For more information see "MagicDraw User Manual" > "Performance Improvement" section. Manual is available in < MagicDraw installation directory > \manual\