AtTask On-premise Optimization Tips

Josh Hardman, Manager, Customer Support, AtTask Inc.

Introduction

This document is intended for administrators hosting their own AtTask server. OnDemand customers already have an optimized environment.

As you may know by now, AtTask runs on a JBoss Application Server. What's so great about this? JBoss Application Server is a community driven project that receives continuous upgrades and improvements. It also runs on Java which makes it very versatile. It should be noted that JBoss is delivered with an AtTask install and does not need to be installed separately. Also know that AtTask cannot be installed into another instance of JBoss outside of the delivered version.

Ok, let's take a look at where the configuration files live and what you can change to optimize your server. Most of the modifications will take place in different locations depending on your server operating system.

Java

You should always maintain an up-to-date JDK. During install, you are prompted to select the Java executable you would like to use (on OS X the installer automatically detects the system default Java). You have the choice of selecting a previously installed Java or Windows and Linux systems will be prompted with a choice of an AtTask delivered version. Please note that the delivered version is a 32bit JDK; a 64bit JDK is supported but must be installed separately. After installation, you may want to change the location in order to update the JDK installed and not necessarily want to re-install AtTask. This can be done with a very simple change. First let's look at this when using a Windows service installation.

Navigate to the folder "AtTask\Windows Service". In this folder open the file "wrapper.conf" in a text editor such as Word Pad. On line 8 you should see the full path to the java executable that you are currently using with AtTask. Simply change the location to the newly installed JDK and you're ready to install your modified AtTask service. Save the file, stop the AtTask service, run the file "UninstallAtTaskService.bat", run the file "InstallAtTaskService.bat" and start the service again. That's it!

If you are running the server on Windows and executing the run.bat file as opposed to the Windows Service option, you will need to modify a separate file. Navigate to the folder "AtTask\AtTaskJBoss\bin" and open the file "run.bat". Line 34 will contain the location of the "JAVA_HOME" variable. Just change this location to the new location, save the file, and restart AtTask.

Now let's look at doing this on Linux/Unix/OS X/etc. Navigate to the folder "AtTask/AtTaskJBoss/bin" and open the file "run.sh" in a text editor. Scroll down to line 56 and you should see the "JAVA_HOME" variable being set to the current JDK location. All you need to do is replace the current path to the path of the new JDK installation, save the file, and restart your AtTask service.

Memory Allocation

Next let's take a look at how you can change the memory allocation without re-installing AtTask. During the AtTask install you are allowed to set your memory allocation, although the maximum is 4GB. Many servers have more memory available and the administrator would like to use it up. Again, let's first look at how to do this when running as a Windows service.

We need to edit the same "wrapper.conf" file from above. The maximum memory setting is located on line 5. It is measured in Mega Bytes and should contain an "m" at the end of the number. For example, "4096m" would represent 4GB (no quotes needed). Just replace the number with the accurate maximum amount of memory you would like to dedicate to the AtTask service. The memory allocated here should be less than the total hardware memory on the server as some memory is needed to run the operating system and other services.

Along with modifying the maximum amount of memory, you may also want to modify the minimum. The minimum amount is reserved at the time the AtTask service is started and the process is allowed to scale up to the maximum amount as needed. Raising the minimum may result in your system doing less work as you use the application. Line 47 contains the setting for the minimum memory. If memory is not a problem for your server, you may want to configure this to match your maximum memory setting.

One final memory optimization that can be made is called the "MaxPermSize". This parameter controls the amount of memory that Java utilizes for garbage collection. My recommendation for this parameter is 256m. 512m may be needed for very large systems running with more than 8GB of memory dedicated to AtTask. Anything above 512m is typically not needed.

After you save the file, follow the same steps as before and uninstall, re-install, and re-start the AtTask service.

The same parameters and options can be found in the corresponding files for running AtTask with the bat file on Windows or the shell scripts on Unix based systems. I will quickly point to their locations.

Windows "run.bat":

Maximum Memory: Line 37 signified by "-Xmx".
Minimum Memory: Line 37 signified by "-Xms".
MaxPermSize: Line 37 signified by "-XX:MaxPermSize".

Linux/Unix/OS X/etc "run.sh":

Maximum Memory: Line 59 signified by "-Xmx".
Minimum Memory: Line 59 signified by "-Xms".
MaxPermSize: Line 59 signified by "-XX:MaxPermSize".

That takes care of the optimizations done at the configuration file level.