# How can I set memory in Java to have better runtime performance?

Run time performance is influenced by memory. Operations made in memory without swapping to disk are better performing. Java tries to gain memory by periodically performing a cleaning procedure called Garbage Collector; during this procedure performance is slower.

Providing more memory to Java and isCOBOL will allow more operations to be made in memory and decrease the number of times Java will need to invoke the Garbage Collector. This will in turn increase runtime performance.

The JVM memory is controlled by five options. The most commonly set are these two:

```
   -Xms        Set initial Java heap size (in bytes)    -Xmx        Set maximum Java heap
size (in bytes)
```

Where available, you should think about using the following Java options:

```
   -server
```

This option is usually available only with the JDK (the JRE doesn't provide it). It causes Java to work in server mode instead of the default client mode. The server mode causes background operations (such as file i/o) to be better performing.

For example, in order to specify a memory limit of 512 MB and the -server option, you use:

```
   iscrun -J-Xmx512m -J-server PROGRAM
```

or if you run using java.exe:

```
   java -Xmx512m -server PROGRAM
```

There is more detailed information about optimizing your runtime performance in the isCOBOL Documentation's Appendices under "Performance Tuning: Guidelines for better runtime performance: Better run time performance"