Increasing JVM maximum memory allocation

The JVM can be configured with a larger heap with the -Xmx option (see oracle docs below). ISymphony allows you to configure any -X option in the jvm.args located in the root of the iSymphony installation directory (default: /opt/isymphony3/server/).

Example: Prepending heap option

```
-Xmx6g
-Xms256m
-XX:MinHeapFreeRatio=30
-XX:MaxHeapFreeRatio=50
-XX:-OmitStackTraceInFastThrow
-Dcom.sun.management.jmxremote.port=8000
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

Oracle JRockit Documentation

From: https://docs.oracle.com/cd/E13150_01/jrockit_jvm/jrockit/jrdocs/refman/optionX.html

-Xmx

This option sets the maximum Java heap size. The Java heap (the “heap”) is the part of the memory where blocks of memory are allocated to objects and freed during garbage collection. Depending upon the kind of operating system you are running, the maximum value you can set for the Java heap can vary.

Note:	`-Xmx` *does not* limit the total amount of memory that the JVM can use.

Operation

Format: `-Xmx<size>[g|G|m|M|k|K]`

Combine `-Xmx` with a memory value

For Example:

`java -Xmx:1g` myApp

sets the maximum java heap to 1 gigabyte.

If you do not add a unit, you will get the exact value you state; for example, 64 will be interpreted as 64 bytes, not 64 megabytes or 64 kilobytes.

The `-Xmx` option and `-Xms` option in combination are used to limit the Java heap size. The Java heap can never grow larger than `-Xmx`. Also, the `-Xms` value can be used as “minimum heap size” to set a fixed heap size by setting `-Xms = -Xmx` when, for example, you want to run benchmark tests.

Known Issue for Linux Users

The JRockit JVM R26.0.0 on Linux IA32 can experience problems setting up memory for object allocation. When this happens, you will receive the following message:

```
[JRockit] ERROR: Fatal error in JRockit during memory setup phase.
Try to reduce the heap size using -Xmx:<size>m, i.e. “-Xmx:16m”. Could
not create the Java virtual machine.
```

and JRockit JVM will be exited.

The workaround for this situation is to try different `-Xmx` values until you find a heap size that is setup correct.

Note:	This known issue is valid for R26.0.0.

Default Value

If you do not set this, the maximum java heap size depends on the platform and the amount of memory in the system as described in Table 2-6.

Table 2-6 Default Maximum Heap Sizes

Release	Platform	Default Maximum Heap Size
R27.2 and older	Windows	75% of total physical memory up to 1 GB
R27.2 and older	Linux, Solaris	50% of available physical memory up to 1 GB
R27.3 and newer	Windows on a 64 bit platform	75% of total physical memory up to 2 GB
R27.3 and newer	Linux or Solaris on a 64 bit platform	50% of available physical memory up to 2 GB
R72.3 and newer	Windows on a 32 bit platform	75% of total physical memory up to 1 GB
R27.3 and newer	Linux on a 32 bit platform	50% of available physical memory up to 1 GB

Flags or Other Options Affected

None.

Exceptions

When using-Xmx, be aware of the following exceptions:

- If both `-Xmx` and `-Xms` are specified the value of `-Xmx` must be larger than or equal to that of `-Xms`.
- If both `-Xmx` and `-Xns` are specified the value of `-Xmx` must be larger than or equal to that of `-Xns`.
- The minimum value for `-Xmx` is 16 MB.