Informatica® Informatica Data Quality 10.0
Informatica® Informatica Data Services 10.0
Informatica® Big Data Management 10.0

Improving the Performance of the Model Repository

Informatica Informatica Data Quality Informatica Data Services Big Data Management Improving the Performance of the Model Repository
10.010.010.0
December 2015

Publication Date: 2021-08-16

Table of Contents

Abstract

The Model repository is a relational database that contains metadata about connections, applications and workflows, transformations and functions, and other objects. You can perform several tasks to improve the performance of the Model repository and its interactions with other Informatica services, and with databases and external clients.

CHAPTER 1

Improving the Performance of the Model Repository

This chapter includes the following topics:

Overview

The Model repository is a relational database instance managed by the Model Repository Service. The database stores tables that hold metadata about repository objects, such as mappings and transformations. Over time, Model repository performance can slow for many reasons. This article describes methods to improve Model repository performance.

You can perform the following tasks to improve the performance of the Model repository:

Follow system installation tips.

When you install various elements of the Informatica domain, you can follow rules and guidelines that lead to good performance.

Manage and purge repository statistics.

As the Data Integration Service runs mappings, workflows, and other processes, statistics accumulate in the Model repository. Manage and purge statistics regularly to maintain Model repository performance.

Manage and purge log entries.

As the Data Integration Service runs mappings, workflows, and other processes, log entries accumulate in the Model repository. Manage and purge log entries regularly to maintain Model repository performance.

Tune the Model repository.

To improve Model repository performance, you can perform several tasks, such as configuring connections and memory settings.

Improve client interactions.

You can configure external settings in clients and databases to improve the speed at which they interact with the Model repository

System Installation Rules and Guidelines

The following rules and guidelines for installation and deployment of Informatica Platform elements pertain to performance optimization:

Host the Model repository database on the same machine as the Model Repository Service.

The Model Repository Service process performance might be slower if a remote server hosts the Model repository database. A high-latency network could slow down the Model Repository Service.

Host the Model repository database on the same network as the Model Repository Service.

If you cannot host the Model repository database on the same machine as the Model Repository Service, host the Model repository database on the same network as the Model Repository Service.

Store monitoring statistics in a separate Model repository

You can set up a separate Model repository to store monitoring statistics. When you store monitoring statistics separately, you control the size of the Model repository that you use for designing mappings, workflows, and applications.

Manage and Purge Repository Statistics

To improve Model repository performance, you can control the amount of statistics that the Model repository retains, and the amount of time that it retains them.

When you configure monitoring in the domain, the Data Integration Services store statistics and reports in a Model repository. As mappings, workflows, and other objects run, the Data Integration Service Manager persists, updates, retrieves, and publishes run-time statistics in the Model repository. The statistics include historical information about objects that the Data Integration Services run. View statistics on the Monitor tab.

To preserve optimal performance, the most important property to set is the Purge Statistics Every property. Set this property to purge the statistics at regular intervals.

For faster performance, you can also limit the number of days that the Model repository preserves summary and detailed historical data and limit the number of sortable historical records.

The following table describes the Monitoring options that you can set and that affect performance:

Option	Recommended Settings for Faster Performance
Preserve Summary Historical Data	Number of days that the Model repository saves averaged data. If purging is disabled, then the Model repository saves the data indefinitely. Default is 180. Minimum is 0. Maximum is 366. For faster performance, reduce the setting to a time period that allows you to continue to access the information you need.
Preserve Detailed Historical Data	Number of days that the Model repository saves per-minute data. If purging is disabled, then the Model repository saves the data indefinitely. Default is 14. Minimum is 1. Maximum is 14. For faster performance, reduce the setting to a time period that allows you to continue to access the information you need.
Purge Statistics Every	Interval, in days, at which the Model Repository Service purges data that is older than the values configured in the **Preserve** options. Default is 1 day. For optimal performance, do not increase the interval.
Days At	Time of day when the Model Repository Service purges statistics. Default is 1:00 a.m. For optimal performance, choose a time when demands on the system are lowest.
Maximum Number of Sortable Records	Maximum number of records that can be sorted in the **Monitor** tab. If the number of records on the **Monitor** tab is greater than this value, then you can only sort by **Start Time** and **End Time**. Default is 3,000. For faster performance, reduce the number to a setting that allows you to continue to access the information you need.
Show Milliseconds	Include milliseconds for date and time fields in the Monitor tab. For faster performance, unselect this option

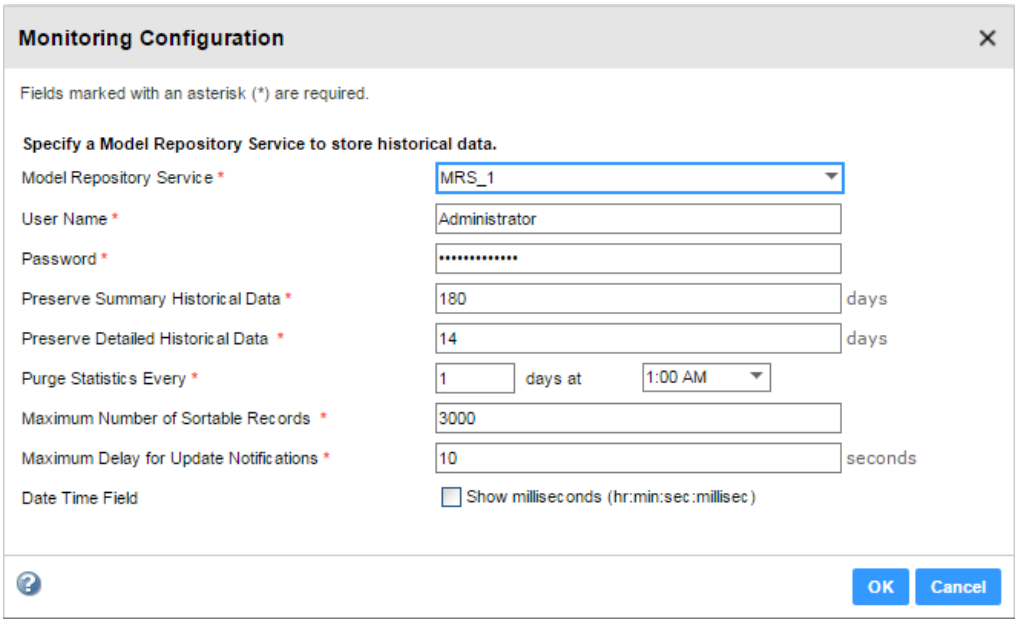
Managing and Purging Statistics

The Model repository saves historical data and sortable records for a period of time that you set, and purges them atintervals that you set.

1. In the Administrator tool, click the **Manage** tab > **Services and Nodes** view.
2. Click the **Monitoring Configuration** tab.

   The current monitoring configuration appears.

3. Click the **Edit** icon to change the monitoring configuration.



The **Monitoring Configuration** window opens.

4. Edit the **Purge Statistics Every** property to purge statistics at regular intervals.
5. Edit the other monitoring properties to preserve optimal performance, click **OK**, and then click **Save**.

To apply the settings, you must recycle all of the Data Integration Services.

Manage and Purge Log Entries

To improve the Model repository performance, you can control the number of log entries that the Model repository retains, and the amount of time that it retains them.

The Data Integration Service Manager accumulates log events for the domain, application services, and users. The log events contain operational and error messages for a domain. The Service Manager and the application services send log events to the Log Manager. When the Log Manager receives log events, it generates log event files. You can view service log events in the Administrator tool.

To preserve optimal performance, you can set the Model repository to control the automatic purge of log events. You can also purge log events manually.

The following table describes the log management options that you can set and that affect performance:

Option	Description
Preserve logs for number of days	Number of days to preserve logs. Default is 30. For faster performance, reduce the number of days to a time period that allows you to continue to access the information you need.
Maximum size for logs in MB	Number of megabytes of disk space to store logs. Default is 200. For faster performance, reduce the amount of storage space that allows you to continue to access the information you need.

Managing and Purging Log Entries

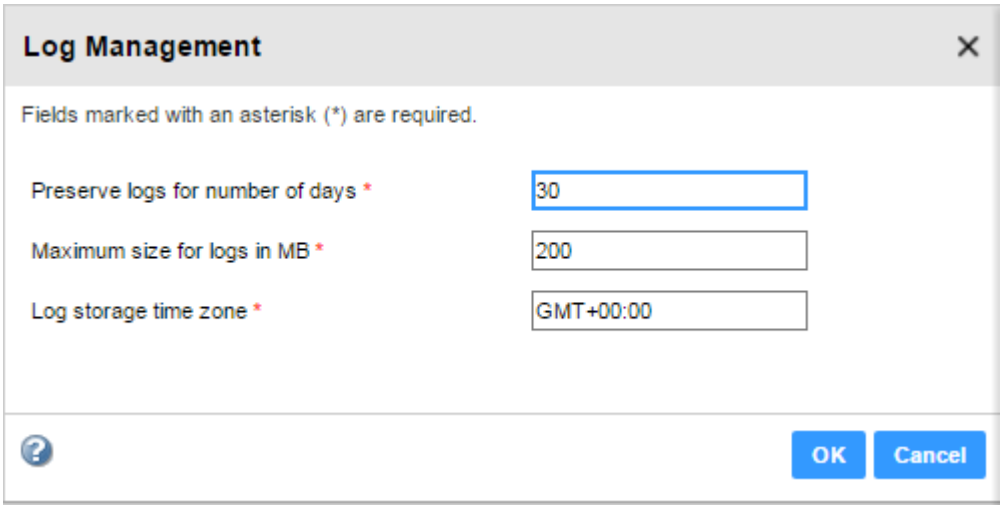
To preserve optimal performance, you can set the Model repository to control the automatic purge of log events.

The Model repository accumulates log events for the domain, application services, and users. The log events contain operational and error messages for a domain.

To preserve optimal performance, limit the size of logs and the number of days to retain them.

1. In the Administrator tool, click the **Logs** tab.
2. Select **Actions** > **Log Management**.

   The **Log Management** dialog box appears.



3. To configure the Model repository to automatically purge log entries more frequently, set one or both of the following properties:
   - Reduce the value of the **Preserve logs for number of days** property.
   - Reduce the value of the **Maximum size for logs in MB** property.
4. Click OK.

Recycle the Model Repository Service for your changes to take effect.

Tune the Model Repository

Heap size and connection pool properties can affect Model repository performance.

The performance effect of tuning the Model repository parameters depends on your environment and circumstances. When a particular setting is causing a bottleneck, then increasing the value of that property might relieve the issue. For example, you can increase the Java heap size to speed performance. However, the Java heap size of the Model repository is limited by the amount of physical memory on a machine, and setting the parameter to exceed the physical memory available could cause a problem. Consult with system administrators when you plan to tune these parameters.

You can tune the following settings to affect the Model repository performance:

- Maximum Heap Size property
- Java Stack Size property
- Memory settings
- Hibernate Connection Pool Size property

Setting the Maximum Heap Size Property

You can increase the Maximum Heap Size property to increase Model repository performance.

Maximum heap size is the amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Model Repository Service. The default value is 1 GB.

You can increase this property to increase the Model repository performance. Generally, Informatica recommends that you set this property at approximately two times the size of the Model repository. For example, if the Model repository consumes 1 GB of disk space, set this property to 2 GB.

You can also increase this property when you have a large pool of concurrent users. Model repository users include users of the Developer tool and the Analyst tool. Because mappings launch the Data Integration Service at run time, and the Data Integration Service connects to the Model repository, you can also include in the user pool mappings that run automatically.

Note: A value that exceeds the physical memory on the machine that hosts the Model repository might cause an issue. Do not exceed the available physical memory.

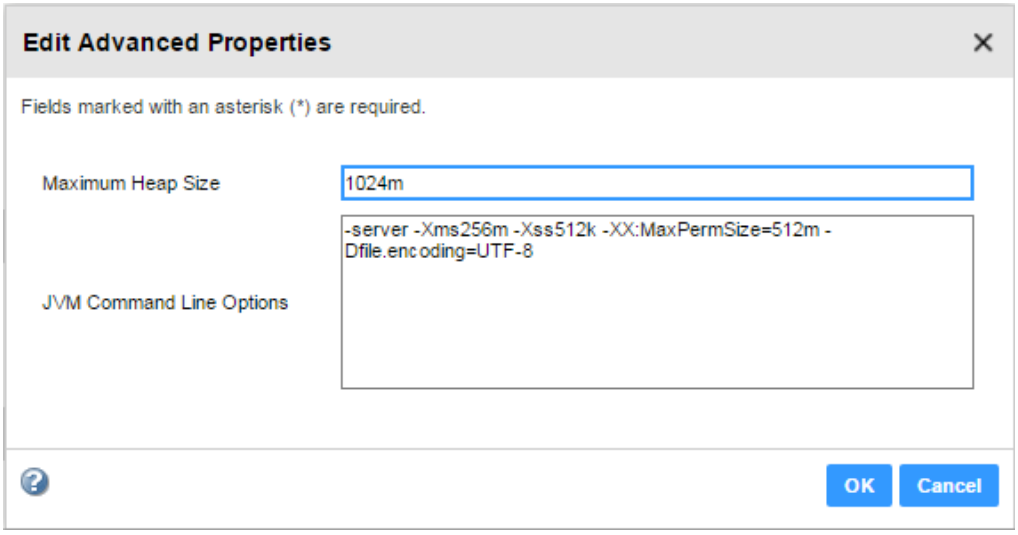
The following table lists guidelines for this property based on the number of concurrent users:

Number of Concurrent Users	Max Heap Size Value
Single user	1 GB
< 10	2 GB
10-50	4 GB
> 50	8 GB

You can consult Informatica Global Customer Support for assistance with this sizing task.

1. In the Administrator tool, click the **Domain** tab.
2. In the **Domain Navigator**, select the Model Repository Service.
3. Click the **Edit** icon in the Advanced Properties section.

The **Edit Advanced Properties** dialog box appears.



4. Increase the **Maximum Heap Size** parameter.
5. Click **OK**.

Recycle the Model Repository Service for your changes to take effect.

Setting the Java Stack Size Property

You can increase the Java Stack Size property to increase Model repository performance.

The Java stack size is the size limit of each Java thread in the Java Virtual Machine (JVM) that runs the Model Repository Service. The default value is 512K.

You can increase this property to make more memory available to Model Repository Service processes.

Note: As with other tunable parameters, you should consult system administrators before changing this setting. The amount of available memory is limited by the physical memory of the machine.

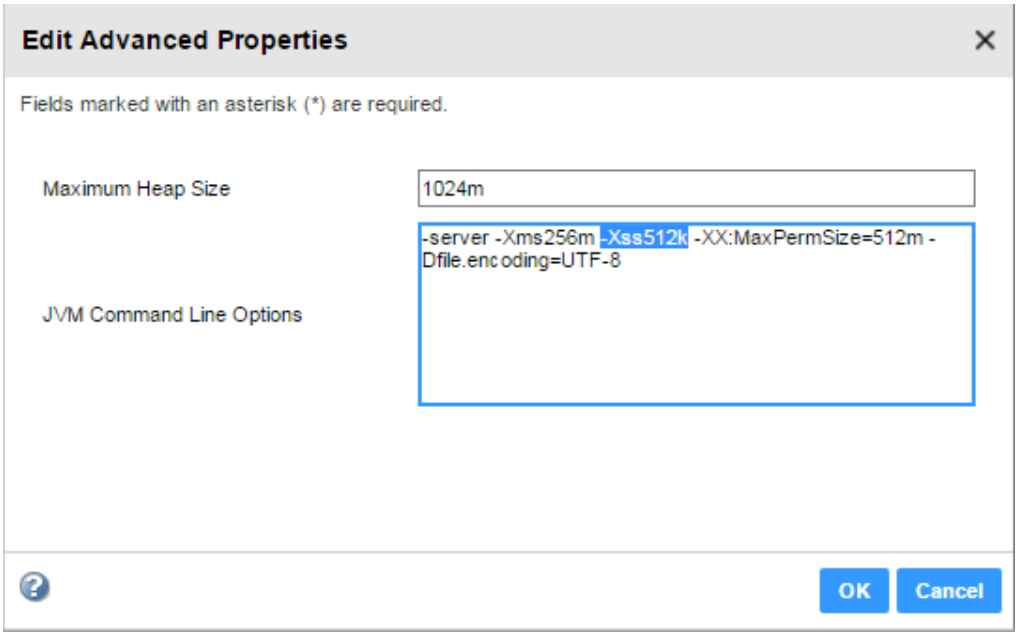
You can consult Informatica Global Customer Support for assistance with this sizing task.

Set the Java stack size property in the **JVM Command Line Options** property in the **Edit Advanced Properties** dialog box. The `-Xss` option expresses the Java stack size.

1. In the Administrator tool, click the **Domain** tab.
2. In the **Domain Navigator**, select the Model Repository Service.
3. Click the **Edit** icon in the Advanced Properties section.

   The **Edit Advanced Properties** dialog box appears.

The following image shows the `-Xss` property highlighted in the **Edit Advanced Properties** dialog box:



4. In the **JVM Command Line Options** window, increase the value of the `-Xss` parameter.
5. Click **OK**.

Recycle the Model Repository Service for your changes to take effect.

Configuring Cache Memory Settings

You can configure the Model repository to use cache memory.

When you configure the Model repository to use cache memory, the Model Repository Service stores objects in memory. Access to objects in memory is faster than access to objects in the repository database.

To configure the amount of memory allocated to cache, you can optionally change the **Cache JVM Options** parameter. This includes the maximum heap size setting, specified by the -Xmx option.

When the number of objects in the Model repository is very large, you can increase the maximum heap size to increase performance. For example, the default property value is `-Xmx128m`. Change the integers to 256 or another multiple of 128 to increase the JVM cache.

However, you should only change this property with the guidance of Informatica Global Customer Support. File a service request for assistance with this issue.

Note: The options you configure in this field do not apply to the JVM that runs the Model Repository Service.

The following image shows the **Edit Cache Properties** dialog box:

Recycle the Model Repository Service for your changes to take effect.

Setting the Hibernate Connection Pool Size Property

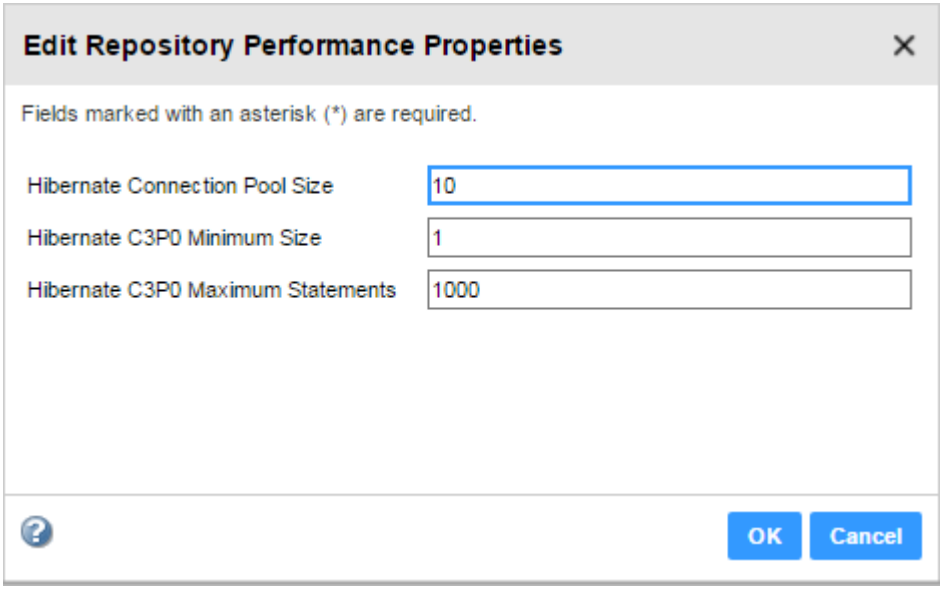
In deployments with many concurrent connections, you can configure connection and statement pooling for the Model repository to increase performance.

The Hibernate Connection Pool Size property establishes the number of connections that are permitted between the Model repository and the Model Repository Service database. The default value is 10.

In deployments with many concurrent connections, you can increase the property to increase performance. However, you should only do this under the guidance of Informatica Global Customer Support. File a service request for assistance with this issue.

1. In the Administrator tool, click the **Domain** tab.
2. In the **Domain Navigator**, select the Model Repository Service.
3. Click the **Processes** view.
4. Click **Edit** in the Repository Performance Properties section.

The **Edit Repository Performance Properties** dialog box appears.



5. Increase the value of the **Cache JVM Options** parameter.
6. Click OK.

Restart the Model Repository Service for your changes to take effect.

Improve Client Interactions

You can configure external settings in clients and databases to improve the speed at which they interact with the Model repository.

To improve Model repository performance, you can perform the following tasks:

Reduce the number of open connections.

When developers have a large number of projects and folders open in the Developer tool, the connection time between the Developer tool and the Model repository might degrade.

To mitigate this issue, instruct developers to open a limited number of projects and folders. Ideally, each developer opens only the project and folder in which they are developing objects, and closes the folder when the development task ends or pauses.

Note: This workaround is not applicable when Informatica is installed in a Citrix virtual environment.

Tune Oracle database settings.

When you use Oracle as the Model repository database, you can tune the open_cursors parameter to increase performance.

The **open_cursors** parameter specifies the number of open cursors for a session. The primary purpose of the parameter is to prevent an excessive number of open cursors.

Cursor use increases when the Data Integration Service deploys applications. When you increase the number of open cursors, you can increase performance.

Contact Informatica Global Customer Support for guidance in increasing the value of this parameter.

Allot additional memory to the Developer tool.

When you deploy applications from the Developer tool, the Deployment Manager fetches all of the objects in the application to Developer tool memory for validation before it deploys the application. When you increase the memory available to this process, the process is faster.

To allot additional memory to this process, edit the **-Xmx** parameter in the Developer tool developerCore.ini file.

Note: The maximum allowed setting is 1500 MB. If you enter a higher setting, the Developer tool will not start.

Allot additional memory to the Data Integration Service process.

The Data Integration Service deploys and runs Model repository objects and applications. When you allot more memory to this process, the process is faster.

To allot additional memory to this process, edit the **Maximum Heap Size** property in the Advanced Properties for the Data Integration Service process.

Allotting Additional Memory to the Developer Tool

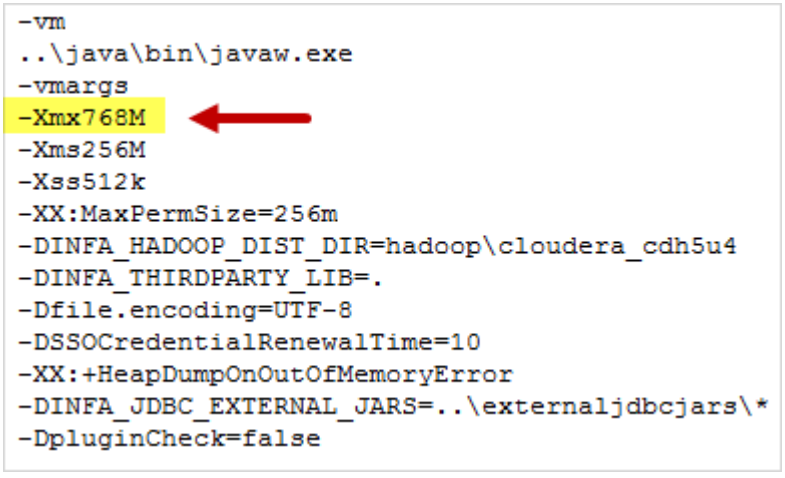
You can increase the maximum Java heap size to allot more memory to the Developer tool. When the Developer tool has more memory to use, it validates applications faster.

1. Browse to the following location:

   `<InformaticaInstallationDir>\clients\DeveloperClient`

2. Edit the DeveloperCore.ini file in a text editor.
3. Edit the line beginning with the characters `-Xmx` to increase the maximum Java heap size.

   For example, to set the Maximum Java heap to 2 GB, edit the parameter to be `-Xmx2G`.
   The following image shows sample text in a DeveloperCore.ini file. Adjust the highlighted memory parameter:



4. Save and close the DeveloperCore.ini file.

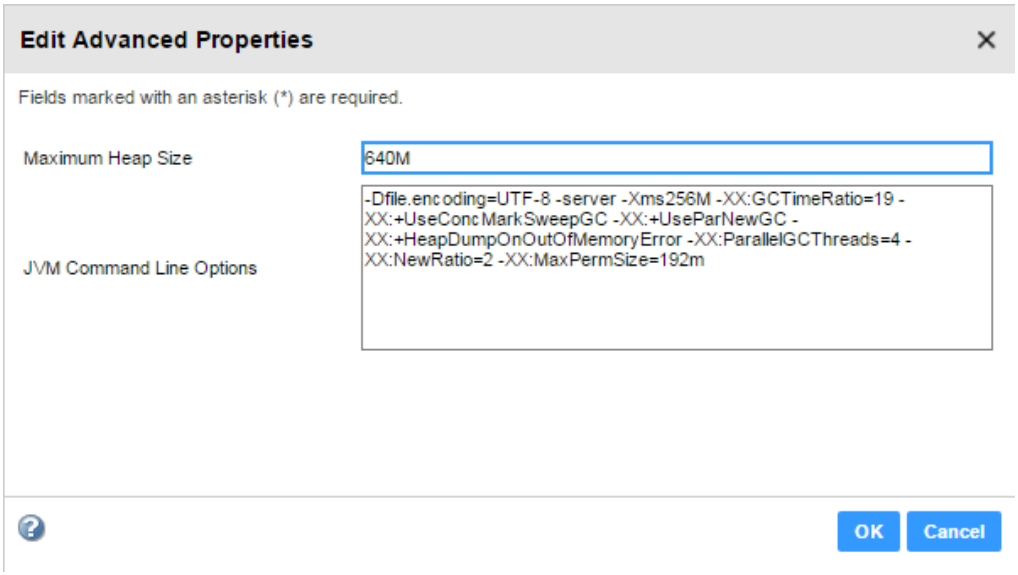
Restart the Developer tool for your changes to take effect.

Allotting Additional Memory to the Data Integration Service Process

You can make more memory available to the Data Integration Service process. When this process has more memory to use, it runs mappings, workflows, and applications faster.

1. In the Administrator tool, click the **Services and Nodes** tab.
2. In the **Domain Navigator**, select the Data Integration Service.
3. Click the **Processes** view.
4. Click **Edit** in the Advanced Properties section.

   The **Edit Advanced Properties** dialog box appears.



5. Increase the value of the **Maximum Heap Size** parameter.
6. Click OK.

Restart the Data integration Service for your changes to take effect.