Cúram 8.1.3

Cúram JMX Configuration Guide

Note

Before using this information and the product it supports, read the information in

Edition

This edition applies to Cúram 8.1, 8.1.1, 8.1.2, and 8.1.3.

© Merative US L.P. 2012, 2024

Merative and the Merative Logo are trademarks of Merative US L.P. in the United States and other countries.

Contents

1 Configuring JMX statistics

Java™ Management Extensions (JMX) is the system that provides operational data from a running application. Configure JMX options to allow operational data from a running application to be collected. This includes invocation targets such as URLs, business methods, and SQL statements. These configuration options are managed by an system administrator in the runtime administration application. There is no software development required to make these configuration changes.

The following is a list of actions that a system administrator can perform that harness the capabilities of JMX:

- Enable the collection of statistics on various invocation targets: URLs, business methods and SQL statements - this can provide a long term view on the performance characteristics of individual invocation targets.
- Enable the collection of statistics on in-flight invocation targets: URLs, business methods and SQL statements - this is helpful in identifying long running or stuck invocation targets. It can also be used to derive a measure of the concurrent activity in the system.
- Enable the collection of aggregated transaction tracing information - this provides detailed aggregated statistics on all call flows executed in the application (from URLs to business methods to SQL statements). This data can be used to trace/profile poorly performing business transactions. In many cases such a snapshot provides enough information for an experienced developer to identify and fix the problem without much further investigations.
- Enable the collection of end user statistics (at the moment these are just browser render times) - this can help in finding performance issues with the application user interface. For instance a custom widget can suddenly start to have performance problems due to inefficient rendering in the browser; using these statistics such a problem can be identified without contacting the affected users or even before they realize that a performance problem is slowly creeping in.
- Download a compressed statistics file containing an XML file with all the JMX statistics from the application and its corresponding XSL style sheet.
- Access the downloaded statistics. For more information, see to view the collected statistics, for example: application URLs, BPOs, SQL, caches, and configuration parameters (properties).

1.1 Configuration parameters

The following is the core set of properties that control the JMX infrastructure in Cúram.

- **curam.jmx.monitoring_enabled**
  A boolean value that enables or disables the JMX monitoring infrastructure.

- **curam.jmx.configured_mbeans_ejb**

  The list of configured MBeans in the EJB container. The syntax is described as follows:

```
Mbean_implementation_class_name1[#MBean_interface_class_name1]
[, Mbean_implementation_class_name2[#MBean_interface_class_name2]...]
```

The content in brackets is optional. If the class name for the interface is not specified it is assumed to be derivable from the implementation class name by appending `MBean` to it.

- **curam.jmx.configured_mbeans_web**
  The list of configured MBeans in the Web container. The syntax is the same as for `curam_jmx_configured_mbeans_ejb`.

- **curam.jmx.per_user_statistics_filter**

  A regular expression that identifies users for which individual statistics are to be collected. This property instructs Cúram MBeans to start aggregating statistics for the set of users that satisfies the regular expression. Custom MBeans can choose to follow the same rule.

  Each user whose user name matches the regular expressions causes a separate set of statistics to be created in memory. Ensure that you have enough memory to store all statistics sets generated by this mechanism.

  > **Note** If the filter used is not very selective ensure you have enough memory allocated in the JVM to store all the generated statistics sets

- **curam.jmx.transaction_tracing_enabled**

  A boolean value that controls whether transaction tracing is enabled in the application. When this boolean is enabled, in flight data collection is also enabled.

  Transaction tracing can require large amounts of memory, so this feature should only be enabled for short period of times for troubleshooting problems. The mechanism can also be fine-tuned to minimize the impact on the application using the other transaction tracing configuration parameters:

  - `curam.jmx.transaction_tracing_max_recorded_threads`
  - `curam.jmx.transaction_tracing_purge_period`
  - `curam.jmx.transaction_tracing_max_thread_idle_time`

  > **Note** Activate transaction tracing only when troubleshooting problems.

  Transaction tracing data can be obtained from the JMX statistics. The data is in XML format, the schema is documented in [XML Schema for Aggregated Transaction Tracing](#).

- **curam.jmx.transaction_tracing_url_filter**
  A regular expression that identifies URLs for which transaction tracing data is collected. This parameter does not impact the run-time aspects of the transaction tracing mechanism; the filter is only applied when the collected data is externalized.

- **curam.jmx.transaction_tracing_max_recorded_threads**

  The maximum number of threads for which transaction tracing data is collected. At any one time there could be more than this number of threads in the transaction tracing data but a significant amount of entries will only be preserved for this number of threads.

  To preserve transaction tracing data for all threads in the application ensure the value of this parameter is equal to or greater than the total number of threads in your application

- **curam.jmx.transaction_tracing_purge_period**

  The period of time, in seconds, between checks to ensure that only the number of threads specified in `curam.jmx.transaction_tracing_max_recorded_threads` are preserved in the transaction tracing data.

- **curam.jmx.transaction _tracing_max_thread_idle_time**

  The maximum amount of time, in seconds, a thread is allowed to be idle before its transaction tracing data can be cleared. This parameter is used to trim data when there are more threads in the application than the value specified by `curam.jmx.transaction _tracing_max_recorded_threads`.

- **curam.jmx.download_statistics_username**

  The username that can download the JMX statistics.

- **curam.jmx.download_statistics_allowed**
  A boolean value that controls access to the JMX statistics.

- **curam.jmx.in_flight_statistics_enabled**
  A boolean value that controls whether statistics about in flight transactions are collected.

- **curam.jmx.in_flight_statistics_custom_handler_class**

  The class name for the custom handler of in-flight statistics. The class must be an implementation of `curam.util.jmx.inflight.InFlightDataCustomHandler`.

- **curam.jmx.in_flight_statistics_custom_handler_enabled**

  A boolean value that controls whether the custom handler for in flight statistics is enabled. If this value is true and no value is specified in `curam.jmx.in_flight_statistics_custom_handler_class` then a built-in handler is activated that outputs the in-flight statistics to the file system. This built-in handler can be configured using the following application properties:

  - *curam.jmx.in_flight_statistics_custom_handler_to_file_output_folder* - the output folder where the default custom handler writes data to files
  - *curam.jmx.in_flight_statistics_custom_handler_to_file_max_file_size* - the maximum file size; when it is reached a new file is created

- **curam.jmx.in_flight_statistics_custom_handler_batch_size**
  The number of in-flight data records buffered before being passed to the custom handler for processing.

- **curam.jmx.in_flight_statistics_custom_handler_enable_data_access_stats**

  A boolean value that when set to false, the in-flight statistics at the data access layer are not forwarded to the custom handler.

- **curam.jmx.output_statistics_timer_enabled**
  A boolean value that controls whether the timer that outputs the JMX statistics is enabled.

- **curam.jmx.output_statistics_timer_period**
  The period at which JMX statistics are output by the timer.

- **curam.jmx.output_statistics_timer_use_delta**
  A boolean value that controls whether the JMX timer resets the statistics after each cycle.

- **curam.jmx.output_statistics_timer_folder**
  The name of a directory on the file system in which JMX reports get created by the JMX timer or Batch Launcher.

- **curam.jmx.output.per.batch**
  A boolean value that controls whether the Batch Launcher will generate a JMX report at the end of each batch program.

- **curam.jmx.timerperiod.batch**
  For batch programs only. Overrides the value specified by property curam.jmx.output_statistics_timer_period.

  Useful for producing intermediate JMX reports for very long running batch programs without having to wait for the program to complete.

  A negative value will disable the JMX timer for batch programs.

1.2 Accessing JMX statistics

You can access the JMX statistics in several ways.

Access JMX statistics in any of the following ways:

- Use a third party monitoring solution - there are enterprise class monitoring solutions with generic JMX support that can be used to visualize the application's statistics.
- Download the statistics as an XML file with an associated XSL file - this method collects the statistics from a single application server and provides a quick way of visualizing the statistics as an HTML page. The drawback is that it only works for application servers hosting the administration application.
- Use the JMX timer mechanism - this method works well in online or batch applications. For online applications, statistics can be configured specifically for each server using JVM property overrides. Timer-based statistics are written discretely for each configured server.

Using the Administration Application

The JMX statistics can be downloaded as a compressed XML file by accessing the URL `/Curam/JMXStats.do` or `/Curam/JMXStats.do?action=download`. The JMX statistics are collected only for the application server that is specified in the URL.

To reset the JMX statistics, invoke `/Curam/JMXStats.do?action=reset`. This invokes the `reset` method on all registered MBeans that define this method, resetting the statistics counters.

The JMX statistics are packaged as an XML file within a ZIP archive file. This archive contains the XML file with the JMX statistics (JMXStats.xml) and an associated XSL style sheet document (JMXStats.xsl). To visualize the statistics unzip the two files on your file system and open the XML file with a browser that supports rendering XML files using XSL style sheets (any modern browser has this capability).

Access to the JMX statistics is controlled using two dynamic application properties:

- `curam.jmx.download_statistics_allowed` - when `true` , access to `/Curam/JMXStats.do` is allowed
- `curam.jmx.download_statistics_username` - this is the username that is allowed to access this URL. If a value is not set for this parameter, access to the JMX statistics is not allowed.

> **Note:** For IBM® WebSphere® Application Server, the "ALL AUTHENTICATED" special subject must be assigned the "Monitor" and "Operator" roles to be able to collect and download Cúram JMX statistics. From the WebSphere® Application Server administrative console click on the Users and Groups menu and from the Administrative group roles add the "Monitor" and "Operator" roles to the special subject "ALL AUTHENTICATED".
>
> A more secure alternative is to assign the "Monitor" and "Operator" roles to the user who is configured to download the statistics. The drawback is that this user bypasses the Cúram login module and it has to be managed inside WebSphere's user registry. Assuming that the configured username is sysadmin, follow these steps:
>
> 1. Add the sysadmin user to the list of excluded usernames for the Cúram login module. For each resource configured with this login module (DEFAULT, RMI_INBOUND and WEB_INBOUND in Global security > JAAS - System logins) click on the `curam.util.security.CuramLoginModule` entry and add to the exclude_usernames list the sysadmin username.
> 2. Add the sysadmin user to the WebSphere's user registry and assign it the "Monitor" and "Operator" roles. In WebSphere administrative console navigate to Users and Groups > Manage Users and define the sysadmin user. Then from Users and Groups > Administrative user roles assign the "Monitor" and "Operator" roles to sysadmin.

To download the JMX statistics, ensure that `curam.jmx.download_statistics_allowed` is set to `true` , log in as the user specified in `curam.jmx.download_statistics_username` and visit the URL `<server>:<port>/Curam/JMXStats.do` or `<server>:<port>/Curam/JMXStats.do?action=download` where `<server>` is the host name of the application server whose JMX statistics are to be downloaded and `<port>` is the port number for your application.

Using the JMX timer

The JMX statistics can be obtained from any environment, online or batch, using the JMX timer. This mechanism writes the JMX statistics periodically at a file system location. The advantage of this mechanism is that it can be used for collecting statistics from every server in an environment.

The JMX timer is controlled by the following application properties:
`curam.jmx.output_statistics_timer_enabled`,
`curam.jmx.output_statistics_timer_period` and
`curam.jmx.output_statistics_timer_use_delta`. For more information on these timers, see [Configuring JMX](#)

1.3 Default application instrumentation

The Cúram application comes with a number of MBeans that expose application statistics.

The MBeans have the following characteristics:

- Some MBeans have their own dynamic configuration data that is accessible with the standard application configuration mechanism.
- Some MBeans gather statistics for users that are selected by using the `curam.jmx.per_user_statistics_filter` parameter.
- The availability of some of the data is affected by either Cúram JMX configuration parameters or the MBean's configuration parameters.

Each of these MBeans is described in the following sections.

CuramServerCoreStatsMBean

This MBean exposes core statistics for code running in the server container.

MBean Configuration Parameters

This MBean has only one parameter:
`curam.jmx.sql_statement_statistics_enabled` - set this parameter to true to enable the collection of SQL statement execution statistics

MBean Statistics

The following statistics are exposed by this MBean.

- Aggregated BPO method execution statistics - the following counters are available:
  - Username – the username that invoked this method. This is only valid when per user statistics are collected.
  - BPO Method – the name of the BPO method
  - Elapsed Time(ms) - the average response time in milliseconds for this BPO method
  - Min Elapsed Time(ms) - the minimum elapsed time in milliseconds
  - Max Elapsed Time(ms) - the maximum elapsed time in milliseconds
  - Std Deviation Elapsed Time(ms) - the standard deviation for the elapsed time
  - Invocations - the number of invocations of this method
  - Errors - the number of exceptions thrown by this method
- Aggregated SQL statement execution statistics - this data is available only if the MBean configuration parameter `curam.jmx.sql_statement_statistics_enabled` is set to `true`.

  - Username – the username that executed this SQL statement. This is only valid when per user statistics are collected.
  - SQL Text – the text of the SQL statement
  - Executions – the total number of times this SQL statement was executed including database and query cache hits

- Elapsed Time(ms) – the average elapsed time in milliseconds to execute the statement. Only occurrences which hit the database are taken into account.
- Min Elapsed Time(ms) - the minimum elapsed time in milliseconds
- Max Elapsed Time(ms) - the maximum elapsed time in milliseconds
- Std Deviation Elapsed Time(ms) - the standard deviation for the elapsed time
- Rows Returned - the average number of rows returned by the statement
- Min Rows Returned - the minimum number of rows returned by the statement
- Max Rows Returned - the maximum number of rows returned by the statement
- Std Deviation Rows Returned - the standard deviation for the number of rows returned by the statement
- Query Cache Hit Ratio - the percentage of query cache hits
- Aggregated application exception statistics

  - Username – the username that invoked the operation that threw the exception. This is only valid when per user statistics are collected.
  - Catalog Entry – the catalog entry for the exception's message
  - Occurrences – the number of occurrences of this exception
- Aggregated SQL query cache statistics

  - Query cache type - the type of the query cache
  - Misses - the number of cache misses
  - Hits – the number of cache hits
- Data about in-flight BPO method and SQL statement invocations - this data is only collected when the Cúram JMX configuration parameter `curam.jmx.in_flight_statistics_enabled` is set to `true`.

  - BPO method data

    - UUID – the unique identifier of this transaction
    - Parent UUID – the unique identifier of this transaction's parent
    - Username – the username that initiated this transaction
    - Method – the method being executed
    - Elapsed Time(ms) - the time elapsed, in milliseconds, since this invocation has been initiated
  - SQL statement data

    - UUID – the unique identifier of this transaction
    - Parent UUID – the unique identifier of this transaction's parent
    - Username – the username that initiated this transaction
    - SQL – the SQL statement being executed
    - Elapsed Time(ms) - the time elapsed, in milliseconds, since this execution has been initiated
- Aggregated transaction tracing statistics - this data is only collected when Cúram JMX configuration parameter `curam.jmx.transaction_tracing_enabled` is set to `true` . The aggregated transaction data is exposed as XML. The XML schema is shown in .

Tracing data is grouped by thread and the meaning of the C element attributes is detailed as follows:

- C – the application layer
- M - the invocation target, depending on the application layer it can be a URL, BPO method or an SQL statement
- EA - the average elapsed time for an invocation
- EMN –the minimum elapsed time for an invocation
- EMX – the maximum elapsed time for an invocation

> **Availability of transaction tracing data** Transaction tracing data is not collected outside JVM process boundaries. This means that full tracing through all application layers is only available when the client and server applications are collocated.

CuramClientCoreStatsMBean

This MBean exposes core statistics for code running in the client container.

MBean Statistics

The following statistics are exposed by this MBean:

- Aggregated URL invocation data - the following counter are available:

  - Username – the username that invoked this URL. This is only valid when per user statistics are collected.
  - URL – the URL for which statistics have been collected
  - Invocations – the number of invocations of this URL
  - Elapsed Time(ms) - The average elapsed time per invocation in milliseconds
  - Min Elapsed Time(ms) - the minimum elapsed time in milliseconds
  - Max Elapsed Time(ms) - the maximum elapsed time in milliseconds
  - Std Deviation Elapsed Time(ms) - the standard deviation for the elapsed time
  - Errors - The number of times an exception is thrown when processing this URL
- Data about in-flight URL invocations - this data is only collected when the Cúram JMX configuration parameter `curam.jmx.in_flight_statistics_enabled` is set to `true`.

  - UUID – the unique identifier of this transaction
  - Parent UUID – the unique identifier of this transaction's parent
  - Username – the username that initiated this transaction
  - URL – the URL being invoked
  - Elapsed Time(ms) - the time elapsed, in milliseconds, since this request has been initiated
  - Client IP address - the IP address of the remote host connected to this application
  - Client port number - the port number of the remote host connected to this application
- Aggregated transaction tracing statistics - this data is only collected when the Cúram JMX configuration parameter `curam.jmx.transaction_tracing_enabled` is set to `true` . If a filter is specified using the configuration parameter

`curam.jmx.transaction_tracing_url_filter`, data is collected only for activity originating from URLs matching the filter. The aggregation data has the same format as for [CuramServerCoreStatsMBean on page 14](#).

CuramCacheStatsMBean

This MBean exposes core statistics for Cúram Cache.

MBean Statistics

The following statistics are exposed by this MBean for each global cache:

- Cache group - the name of the cache group
- Cache – the name of the cache
- Layer – the name of the cache layer (memory, disk,...)
- Size – the number of items in the cache
- Hits - the number of requests to the cache that returned an item already loaded in the cache
- Misses - the number of requests to the cache that returned an item which had to be loaded in the cache
- Evictions - the number of times items that have been evicted from the cache
- Last access time - the timestamp of the last access to the cache
- Average get time(ns) - the average elapsed time, in nanoseconds, that takes for an item to be read from the cache. Note that some cache providers might only support millisecond resolution.
- Min get time(ns) - the minimum get time in nanoseconds
- Max get time(ns) - the maximum get time in nanoseconds
- Std deviation get time(ns) - the standard deviation for the get time

Multi-instance caches have three types of statistics. Some multi-instance caches offer also a statistics differentiator token which is used to differentiate further the cache instances for the purpose of calculating statistics. For transaction local caches this differentiator is the name of the business method currently being executed.

- Snapshot - these are statistics collected at the moment of the query from all caches alive at that moment
- Aggregated - these are statistics that are collected from all instances of caches that have been created. There are two types of aggregated statistics:
  - Without differentiator - aggregated statistics where the differentiator token provided by the cache instance is not taken into consideration
  - With differentiator - aggregated statistics where the differentiator token provided by the cache instance is taken into consideration

CuramEndUserStatsMBean

This MBean exposes end user browser render time statistics.

- URL - the URL for which the statistics apply. The URL may have a list of widget IDs appended to it. These are the widgets that triggered loading events on the page at this URL.

- Invocations - the number of invocations to this URL
- RenderTime1(ms) - render time 1 in milliseconds - this is the time between the browser receiving the <head> tag and the browser receiving the </body> tag. It is an approximation of the time it takes to transfer the HTML document.
- RenderTime2(ms) - render time 2 in milliseconds - this is the time between render timestamp 1 and the time Dojo finished loading. This is an approximation of the time needed by Dojo initialization.
- RenderTime3(ms) - render time 3 in milliseconds - this is the time between render timestamp 2 and the time the last Dojo widget on the page finished loading

> **Performance impact** Enabling the collection of end user statistics may have an impact on the performance of the application.

MBean configuration parameters

The collection of end user statistics is controlled by the following application configuration parameters:

- `curam.jmx.end_user_statistics_enabled` - enables and disables the collection of end user statistics
- `curam.jmx.end_user_statistics_user_filter` - regular expression that selects the users for which end user statistics are collected
- `curam.jmx.end_user_statistics_display_enabled` - whether or not the end user statistics are displayed in the browser. If true, the statistics for the current page are displayed.
- `curam.jmx.end_user_statistics_upload_delay` - the delay in seconds between the page reporting being loaded and the moment the statistics are uploaded

CuramConfigurationStatsMBean

This MBean exposes application configuration parameter statistics.

- Parameter - the name of the runtime configuration parameter
- Type - the type for the value of this parameter
- Value - the current value of this parameter
- Access count - the number of times the configuration parameter was accessed

ExtensionPointsStatsMBean

This MBean exposes execution statistics for instrumented application extension points.

- Extension point - the name of the application extension point
- Invocations - the number of times the extension point was executed
- Errors - the number of times the execution of the extension point failed
- Elapsed time(ms) - the average elapsed time in milliseconds for an execution of the extension point
- Min elapsed time(ms) - the minimum elapsed time in milliseconds

- Max elapsed time(ms) - the maximum elapsed time in milliseconds
- Std deviation elapsed time(ms) - the standard deviation for the elapsed time

OperationalStatsMBean

This MBean provides operational information.

MBean statistics

The following statistics are provided by this MBean:

- Last reset time - the last time the JMX statistics were reset.
- Last download time - the last time the JMX statistics were downloaded
- Application start up time - the time when the application was started

RunAndRecordStatsMBean

This MBean provides statistics for code instrumented with
`CuramJMXUtil.runAndRecord.`.

MBean statistics

The following statistics are provided by this MBean:

- **Target** - the monitored target as provided to the call to `CuramJMXUtil.runAndRecord`
- **Invocations** - the number of invocations made to the monitored target
- **Elapsed time(ms)** - the average elapsed time in milliseconds for an invocation of the monitored target
- **Std deviation elapsed time(ms)** - the standard deviation of the elapsed time in milliseconds
- **Min elapsed time(ms)** - the minimum elapsed time in milliseconds
- **Max elapsed time(ms)** - the maximum elapsed time in milliseconds
- **Errors** - the number of times the invocation failed

CuramWebServicesOutboundPoolStatsMBean

This MBean provides statistics for the pool of connection to external web services.

MBean Statistics

The following statistics are provided by this MBean:

- **Key** - a text representation of the external web service
- **Active** - the number of active connections in the pool
- **Size** - the total number of connections in the pool

CuramWebServicesOutboundStats

This MBean provides statistics for the outbound web service calls.

MBean Statistics

The following statistics are provided by this MBean:

- **Target** - a text representation of the external web service
- **Invocations** - the number of invocations made to the monitored target
- **Elapsed time(ms)** - the average elapsed time in milliseconds for an invocation of the monitored target
- **Std deviation elapsed time(ms)** - the standard deviation of the elapsed time in milliseconds
- **Min elapsed time(ms)** - the minimum elapsed time in milliseconds
- **Max elapsed time(ms)** - the maximum elapsed time in milliseconds
- **Errors** - the number of times the invocation failed

AssessmentEngineStatsMBean

The MBean exposes core statistics from the CER Assessment Engine

Reassessment group stats

Contains the timing information from CER case recalculations banded into ranges.

Reassessment aggregation stats

Reassessment aggregation aggregates multiple reassessment requests into a single reassessment where possible, and prevents contention among requests to reassess the same case. This comprises three steps per case - claimQueue, reassessCase, releaseQueue and this section of JMX report contains commonly used counts and timings for these steps.

CEREngineStatsMBean

The MBean exposes core statistics for the CER Engine.

DependencyManagerStats

This section contains the storing and outdating of dependencies. Storing dependencies for a dependent involves three steps:

1. Reading existing records for dependent (DependencyManager.storeDependencies - Read)
2. Inserting new records if required (DependencyManager.store storeDependencies.batchinsert - Insert)
3. Deleting any unused records (DependencyManager.executedDelayedDeletionsByPK - Delete).

Statistics for steps 1 and 2 are shown grouped by dependent type.

Step 3 is done at the end of each transaction and hence cannot be grouped by dependent type.

Dependency records can also be deleted by dependent type and statistics for this operation are shown grouped by dependent type (DependencyManager.executeDelayedDeletions.removebyPK - Delete).

Outdating dependencies is the process of executing the handler for the dependent type and the statistics for this operation are shown grouped by dependent type (DependencyManager.outdate -).

The sections FinancialComponentStats , PerformDeferredRecalculationStats and ReassessmentStats show commonly used counts and timings for financial component generation, dependency manager recalculations, and CER case recalculations.

1.4 XML Schema for Aggregated Transaction Tracing

```xml
<xsd:schema attributeFormDefault="unqualified"
        elementFormDefault="qualified" version="1.0"
                    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="CURAM_TRANSACTION_TRACE">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" name="THREAD">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element maxOccurs="unbounded" name="C">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" name="C">
                      <xsd:complexType>
                        <xsd:attribute name="C"
                                        type="xsd:string" />
                        <xsd:attribute name="M"
                                        type="xsd:string" />
                        <xsd:attribute name="EA"
                                        type="xsd:decimal" />
                        <xsd:attribute name="EMN"
                                        type="xsd:decimal" />
                        <xsd:attribute name="EMX"
                                        type="xsd:decimal" />
                        <xsd:attribute name="CT"
                                        type="xsd:int" />
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                  <xsd:attribute name="C" type="xsd:string" />
                  <xsd:attribute name="M" type="xsd:string" />
                  <xsd:attribute name="EA" type="xsd:decimal" />
                  <xsd:attribute name="EMN" type="xsd:decimal" />
                  <xsd:attribute name="EMX" type="xsd:decimal" />
                  <xsd:attribute name="CT" type="xsd:int" />
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
            <xsd:attribute name="ID" type="xsd:string" />
          </xsd:complexType>
```

```
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
```

Notices

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the Merative website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of Merative

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Merative.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Merative reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by Merative, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

MERATIVE MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents.

Information concerning non-Merative products was obtained from the suppliers of those products, their published announcements or other publicly available sources. Merative has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-Merative products. Questions on the capabilities of non-Merative products should be addressed to the suppliers of those products.

Any references in this information to non-Merative websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those

websites are not part of the materials for this Merative product and use of those websites is at your own risk.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to Merative, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. Merative, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. Merative shall not be liable for any damages arising out of your use of the sample programs.

Privacy policy

The Merative privacy policy is available at https://www.merative.com/privacy.

Trademarks

Merative ™ and the Merative ™ logo are trademarks of Merative US L.P. in the United States and other countries.

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Adobe™, the Adobe™ logo, PostScript™, and the PostScript™ logo are either registered trademarks or trademarks of Adobe™ Systems Incorporated in the United States, and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft™, Windows™, and the Windows™ logo are trademarks of Microsoft™ Corporation in the United States, other countries, or both.

UNIX™ is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.