


Automated Laboratory Processes

CCLAS 6



Release Notes Addendum

Version 2024-r2

© Copyright 2024 Datamine Software

All Rights Reserved Confidential and Proprietary

Published: Thursday, 12 December 2024

Legal Disclaimer

The product described in this documentation may be connected to, and/or communicate information and data via, a network interface, which should be connected to a secure network. It is your sole responsibility to ensure a secure connection to the network and to establish and maintain appropriate measures (such as but not limited to the installation of firewalls, application of authentication measures, encryption of data, installation of antivirus programs, etc.) to protect the product, the network, your systems, and the interface against any kind of security breach, unauthorised access, interference, intrusion, leakage, damage, or corruption or theft of data. We are not liable for damages or losses related to any such security breach, unauthorised access, interference, intrusion, leakage, damage, or corruption or theft of data.

Contents

CCLAS 6 2024-r2 Release Notes—Addendum	4
2024-r2 base	4

CCLAS 6 2024-r2 Release Notes—Addendum

2024-r2 base

This addendum contains various issues with their accompanying investigations and outcomes regarding deployment, data accessibility, batch job processing, system logging, scripting and development.

Deployment

CCLAS-6210 Audit Azure dependencies expiring by July 2025

Issue

Any dependencies that CCLAS 6 uses to deploy the application to the Azure Cloud should be of the latest versions.

Investigation

Look into the Azure timeline of dependencies and find any forced upgrades.

Outcome

The Azure advertisement of retirements up to July 2025 (Azure Deprecations Timeboard) was not found to include any services that are currently involved in the CCLAS 6 build or runtime. Therefore, no action was found to be required in the CCLAS 6 code or runtime definitions.

System

CCLAS-6221 Investigate Garbage Collector optimisation

Issue

Effective garbage collection tuning is crucial for optimising Java application performance. Tuning is the process of adjusting the startup parameters of your JVM-based application to match the desired results. The right configuration can reduce pause times and improve throughput, significantly impacting your application's efficiency. Therefore, it is important to investigate Garbage Collector (GC) performance in the CCLAS 6 application.

Investigation

Investigate GC optimisation via the available parameters: heap size, alternate GCs, young gen size, time ratio, pause time.

Steps:

1. Perform a 1-hour Benchmark test and record jvm memory behaviour.
2. Research the settings and tune GC accordingly.
3. Perform another 1-hour Benchmark test and record jvm memory behaviour.
4. Demonstrate "better" memory usage due to the revised GC settings.

Outcome

After extensive benchmark testing and analysis, the GC settings have been finalised to optimise the performance of the CCLAS 6 application for both web and batch server environments.

The G1 GC has been chosen for its ability to handle large heaps efficiently and minimise pause times, making it suitable for the application's high-performance and low-latency requirements.

The settings for the Web Server prioritise low latency to ensure responsive interactions for end-users. On the other hand, the Batch Server configuration is designed for throughput, accommodating the higher computational demands of batch processing tasks.

The following JVM options are added to the yaml file for the server startup configuration.

Web Server:

- **-XX:+UseG1GC**
Enables the G1 Garbage Collector.
- **-XX:MaxGCPauseMillis=75**
Modified: Default is 200 ms.
Sets the target maximum garbage collection pause time of 75 ms for better responsiveness.
- **-XX:G1HeapRegionSize=4m**
Modified: Default is determined by the heap size (1 MB to 32 MB).
Configures G1's heap region size to 4 MB for better granularity.
- **-XX:InitiatingHeapOccupancyPercent=40**
Modified: Default is 45%.
Triggers concurrent marking when heap occupancy reaches 40%.
- **-XX:ParallelGCThreads=24**

Modified: Default is the number of available processors.

Allocates 24 threads for parallel GC operations.

- **-XX:ConcGCThreads=18**

Modified: Default is approximately 1/4 of ParallelGCThreads.

Allocates 18 threads for concurrent garbage collection.

- **-XX:+G1UseAdaptiveIHOP**

Dynamically adjusts the initiating heap occupancy percentage.

- **-XX:+UseDynamicNumberOfGCThreads**

Dynamically adjusts the number of GC threads based on workload.

- **-XX:+UseStringDeduplication**

Reduces memory usage by deduplicating identical strings.

Batch Server:

- **-XX:+UseG1GC**

Enables G1 Garbage Collector.

- **-XX:MaxGCPauseMillis=100**

Modified: Default is 200 ms.

Sets the target maximum pause time of 100 ms for garbage collection.

- **-XX:G1HeapRegionSize=4m**

Modified: Default is determined by the heap size (1 MB to 32 MB).

Configures G1's heap region size to 4 MB for better granularity.

- **-XX:InitiatingHeapOccupancyPercent=40**

Modified: Default is 45%.

Triggers concurrent marking when heap occupancy reaches 40%.

- **-XX:ParallelGCThreads=24**

Modified: Default is the number of available processors.

Allocates 24 threads for parallel GC operations.

- **-XX:ConcGCThreads=12**

Modified: Default is approximately 1/4 of ParallelGCThreads.

Allocates 12 threads for concurrent garbage collection.

- **-XX:+G1UseAdaptiveIHOP**

Dynamically adjusts the initiating heap occupancy percentage.

- **-XX:+UseDynamicNumberOfGCThreads**

Dynamically adjusts the number of GC threads based on workload.

- **-XX:+UseStringDeduplication**

Reduces memory usage by deduplicating identical strings.

These configurations balance the unique demands of the web and batch environments while leveraging the flexibility and adaptability of the G1 GC. They provide a robust foundation for efficient memory management, ensuring the CCLAS 6 application performs reliably under varying workloads.

Data Accessibility

CCLAS-5454 Change Libman to do a bulk delete and insert rather than line by line updates

Issue

The LibMan Upload mechanism does massive record removals and insertions. The aim is to see if this process could be sped up.

Investigation

The LibMan Upload mechanism utilises DBUnit tools to do massive record removals and insertions. An investigation was made into whether another tool would be better, however, it was determined that a replacement for DBUnit is not possible at this point because the other component for export heavily relies on the DBUnit XML format to perform data export. An attempt to introduce Camel as a partial processor in the upload process is blocked by the fact that DBUnit cannot share the session together with Camel since Camel has its own Transaction context.

Therefore, for this release, SQL optimisation is the focus.

Outcome

Change the SQL to delete scheme version analyte list items from the CCSHEMEVERANALYTELISTITEM table to leverage a bulk delete, based on partial business keys (ORGANISATION_CODE, SCHEME_LABORATORY_CODE, SCHEME_CODE) instead of being based on the scheme version analyte's ID.

After importing schemes, products and biofields, if the line of business in the source and target package are the same, then the process now skips updating the line of business on those destination records.

CCLAS-6061 Review the Archiving Scripts to see if chunking of update statements to SSA, SSAR and JobAudit is more performant

Issue

The original groovy Archiving Scripts updates ALL records in a partitioned table to the new ArchiveStatus in a single Update Service, for a single Job. The aim is to see if this process could be sped up.

Investigation

Where there are a large number of records to update, performing the updates in chunks, so that each update is for a smaller number of records, can be more performant.

The tables involved in large updates are CCSAMPLESCHEMEANALYTE (SSA), CCSAMPLESCHEMEANALYTEREAD (SSAR) and CCJOBAUDIT

Outcome

The archiving scripts now have in-built chunking logic for updates to the CCSAMPLESCHEMEANALYTE (SSA), CCSAMPLESCHEMEANALYTEREAD (SSAR) and CCJOBAUDIT tables, to make the process more performant.

CCLAS-6193 Move archived records into less costly Oracle spaces

Issue

Investigate whether partitioned tables and Indexes can be moved into a slower/cheaper table space (and disk space) to optimise performance on the daily transactions on the Oracle database.

Investigation

Request advice from **Fix My Data** Oracle experts as to how to move some partitioned tables and indexes into other resources.

Outcome

Note: All this advice is of a general nature and should be analysed and reviewed by the corporate Oracle Database Advisors and adjusted as required before being applied. These changes should be applied to a non-Production copy and tested thoroughly (including the load/stress testing) before attempting to make these changes to production environments.

Advice from Fix My Data:

1. Identify all sub-partitions within each main laboratory partition that needs to be moved to a slower or more cost-effective drive.
2. Create a tablespace (or as many as desired) for the new moved sub-partitions, ensuring each tablespace has its data file located on the new drive.
3. Next, alter the table, moving each sub-partition (keeping it online) to the newly created tablespace, while ensuring the indexes are updated as well.

Important: In order to complete this work, the Oracle DBA Engineer should first backup the database so that it can be restored easily, if required, then carry these steps out in a test environment.

Input to Chat GTP: **When creating a new table space in Oracle 19.3, how to nominate the file disk that should be used to hold the content for that table space?**

Output:

When creating a new tablespace in Oracle 19.3, you can specify the exact file location (disk path) for the tablespace's datafile. This is done by setting the DATAFILE parameter in the CREATE TABLESPACE statement and providing the full path to the file on the desired disk.

Here's a basic example showing how to create a tablespace and specify the location on a particular disk:

Create a table space and specify the location on a particular disk

```
CREATE TABLESPACE archive_tablespace

    DATAFILE '/path_to_disk/your_directory/archive_tablespace01.dbf'

    SIZE 500M

    AUTOEXTEND ON

    NEXT 100M

    MAXSIZE 2G;
```

Explanation of Parameters

- **DATAFILE:** Specifies the full path and filename for the tablespace's datafile. Here, /path_to_disk/your_directory/ would be replaced with the directory path on the desired disk.
- **SIZE:** Sets the initial size of the datafile (for example, 500M).
- **AUTOEXTEND:** Enables auto-extension of the datafile as it fills up, useful for managing tablespace growth.
- **NEXT:** Defines the increment size for each auto-extension (for example, 100M).
- **MAXSIZE:** Sets the maximum size the datafile can grow to (for example, 2G).

Additional Notes

1. **Directory Structure:** Ensure that the directory (/path_to_disk/your_directory/) is created and accessible by the Oracle user.
2. **Permissions:** The Oracle user must have write permissions to the specified disk path.
3. **Multiple Datafiles:** For larger tablespaces, you can add multiple datafiles in the same tablespace, each potentially on different disks:

Add multiple datafiles in the same tablespace

```
CREATE TABLESPACE archive_tablespace  
  
    DATAFILE '/disk1_path/archive_tablespace01.dbf' SIZE 500M,  
  
    '/disk2_path/archive_tablespace02.dbf' SIZE 500M;
```

This way, Oracle stores tablespace data in the specific file paths, enabling control over disk usage and performance optimisation.

Input to Chat GTP: **What oracle 19.3 commands will be required to alter a table, moving each sub-partition (while keeping it online) to a newly created tablespace, while ensuring the indexes are updated as well?**

Output:

To move each sub-partition of a table to a new tablespace in Oracle 19.3 while keeping it online and ensuring the indexes are updated, you'll want to follow these general steps. This process involves altering each sub-partition individually, specifying the new tablespace, and also moving the corresponding index sub-partitions. Here's how to do it:

1. **Identify the table, partitions, and sub-partitions:** Before you proceed, identify the table and its sub-partitions. This can be done with a query on the ALL_TAB_SUBPARTITIONS and ALL_PART_INDEXES views.
2. **Create the new tablespace:** Ensure that your new tablespace is created and has adequate space for the sub-partitions.
3. **Move each sub-partition:** For each sub-partition, use the ALTER TABLE command to move it to the new tablespace. This can be done online by specifying ONLINE.
4. **Update index sub-partitions:** After moving each table sub-partition, move the corresponding index sub-partition(s) as well to the new tablespace.

Here's a general approach using SQL commands:

1. Move table sub-partitions

Move table sub-partitions

```
BEGIN

    FOR sub_part IN (SELECT TABLE_NAME, SUBPARTITION_NAME FROM ALL_TAB_SUBPARTITIONS
                     WHERE TABLE_NAME = 'YOUR_TABLE') LOOP

        EXECUTE IMMEDIATE 'ALTER TABLE ' || sub_part.TABLE_NAME || ' MOVE
                           SUBPARTITION ' || sub_part.SUBPARTITION_NAME ||

                               ' TABLESPACE NEW_TABLESPACE_NAME ONLINE';

    END LOOP;

END;

/
```

The ALTER TABLE ... MOVE SUBPARTITION ... TABLESPACE ... ONLINE command moves each sub-partition to the specified tablespace while keeping the table online, allowing other operations to continue.

2. Move index sub-partitions

Move index sub-partitions

```
BEGIN

    FOR idx_sub_part IN (SELECT INDEX_NAME, SUBPARTITION_NAME FROM ALL_IND_
                        SUBPARTITIONS WHERE INDEX_NAME = 'YOUR_INDEX_NAME') LOOP

        EXECUTE IMMEDIATE 'ALTER INDEX ' || idx_sub_part.INDEX_NAME || ' MODIFY
                           PARTITION ' || idx_sub_part.SUBPARTITION_NAME ||

                               ' TABLESPACE NEW_TABLESPACE_NAME ONLINE';

    END LOOP;

END;

/
```

The ALTER INDEX ... MODIFY PARTITION ... TABLESPACE ... ONLINE command updates each index sub-partition to reflect the new tablespace. Keeping it online allows minimal downtime.

3. Loops for Automation

By iterating over the sub-partitions of both the table and indexes, you ensure each partition is handled without manually specifying every sub-partition name, making the process efficient for tables with numerous sub-partitions.

Important Considerations:

- **Ensure Space Availability:** Before performing the move, ensure that the target tablespace has sufficient space to accommodate the data.
- **Rebuild Indexes if Necessary:** If you encounter performance issues post-move, you may consider rebuilding the indexes.
- **Permissions:** Make sure you have the necessary permissions to alter tables and indexes within the database.

This approach ensures that all sub-partitions of both the table and indexes are moved efficiently and remain online.

CCLAS-6218 Review and amend index use for report request tables

Issue

Updating indexes on a database insert, update or delete can be time consuming, therefore, a review of index use for database reads to report request tables is important so that any redundant indexes can be removed.

Investigation

Index use monitoring was turned on for report request tables for customer and in-house functional and performance test databases to determine which indexes were not used during the monitoring period.

The tables covered by this monitoring were:

- CCREPREQ
- CCREPREQJOB
- CCREPREQSCHEME
- CCREPREQSCHEMEANALYTE
- CCREPREQSAMPLE
- CCREPREQSAMPLESCHEME
- CCREPREQSAMPLESCHEMEANA
- CCREPREQREPORT

The steps used to determine unused indexes are:

- Collect all indexes
- Exclude an index from being dropped if:

- It is frequently used (either by customer or in-house functional and performance test databases, or in code)
- If it is a primary or foreign key
- If it is used as a paging restart index
- No need to drop an index if it is for I18N.

Outcome

After assessing index use, the following indexes are removed to improve performance of database operations:

- CCREPREQJOBTEMPBII1
- CCREPREQSAMPLESCHEMEANABII12
- CCREPREQSCHEMEANALYTE01I1
- CCREPREQSCHEMEANALYTEBII7

Functional and performance tests were run to ensure no degradation of performance.

CCLAS-6219 Build business keys to improve reporting performance

Issue

Business keys often improve database query performance, however, some frequently accessed tables are linked only using surrogate IDs.

Investigation

Joins to the CCREPREQREPORT table are made using the REP_REQ_ID surrogate ID.

Outcome

LABORATORY_CODE, REP_REQ_CODE and ORGANISATION_CODE columns are added to the CCREPREQREPORT table. The values for these fields are backfilled during installation of the release.

The following indexes are added or amended to use these business keys:

Table	Index change	Columns
CCREPREQ	CCREPREQBII5 added; non-unique	LABORATORY_CODE, CODE, ORGANISATION_CODE
CCREPREQJOB	CCREPREQJOB BII3 amended; non-unique	LABORATORY_CODE, REP_REQ_CODE, ORGANISATION_CODE, JOB_CODE
CCREPREQSCHEME	CCREPREQSCHEME	LABORATORY_CODE,

Table	Index change	Columns
	BII7 amended; non-unique	REP_REQ_CODE, ORGANISATION_CODE, JOB_CODE, SCHEME_LABORATORY_CODE, SCHEME_CODE, SCHEME_VERSION_NUMBER
CCREPREQSCHEME ANALYTE	CCREPREQSCHEME ANALYTEBII6 amended; non-unique	LABORATORY_CODE, REP_REQ_CODE, ORGANISATION_CODE, JOB_CODE, SCHEME_LABORATORY_CODE, SCHEME_CODE, SCHEME_VERSION_NUMBER, ANALYTE_CODE
CCREPREQSAMPLE	CCREPREQSAMPLE BII4 amended; non-unique	LABORATORY_CODE, REP_REQ_CODE, ORGANISATION_CODE, JOB_CODE, SAMPLE_CODE
CCREPREQSAMPLE SCHEME	CCREPREQSAMPLE SCHEMEBII5 amended; non-unique	LABORATORY_CODE, REP_REQ_CODE, ORGANISATION_CODE, JOB_CODE, SAMPLE_CODE, SCHEME_LABORATORY_CODE, SCHEME_CODE, SCHEME_VERSION_NUMBER
CCREPREQSAMPLE SCHEMEANA	CCREPREQSAMPLE SCHEMEANABII6 amended; non-unique	LABORATORY_CODE, REP_REQ_CODE, ORGANISATION_CODE, JOB_CODE, SAMPLE_CODE, SCHEME_LABORATORY_CODE, SCHEME_CODE, SCHEME_VERSION_NUMBER, ANALYTE_CODE
CCREPREQREPORT	CCREPREQREPORTBII2 added; non-unique	LABORATORY_CODE, REP_REQ_CODE, ORGANISATION_CODE, REPORT_TEMPLATE_ID, OUTPUT_FORMAT

Note: The indexes are created as non-unique because it was determined in testing that there can be more than one record in any of these report request tables with the same index, caused by the Workbook RR Builder API and Add Batch QC function in Report Requests creating records where the JobCode is not the correct value for the JobID property. These partitioned indexes will be made unique in a future release when the JobCode issue is amended.

Database queries are adjusted to use LABORATORY_CODE and REP_REQ_CODE business keys to enhance efficiency. ORGANISATION_CODE is not currently used queries, but is added for future use in core code or for immediate use in custom scripts.

CCLAS-6220 Apply table and index partitioning for reporting

Issue

The time it takes to generate a job report needs to be quicker since report generation is a frequent process in a laboratory and the system batch processor can become very busy.

Investigation

The tables associated with report requests are not partitioned. Tables associated with jobs and samples are partitioned by LABORATORY_CODE and sub-partitioned by ARCHIVE_STATUS, however, a report request can contain multiple jobs, and at the moment, it cannot be assured that all jobs in a report request have the same archive status, however, it can be ensured that all jobs in a report request come from the same laboratory.

Outcome

To enhance database query efficiency for reporting, report request tables and indexes are partitioned by LABORATORY_CODE. These tables include:

- CCREPREQ
- CCREPREQJOB
- CCREPREQSCHEME
- CCREPREQSCHEMEANALYTE
- CCREPREQSAMPLE
- CCREPREQSAMPLESCHEME
- CCREPREQSAMPLESCHEMEANA
- CCREPREQREPORT

For this work, the following fields were added to the CCREPREQREPORT table:

- REP_REQ_CODE (for query performance, to remove the need to have a table join)
- LABORATORY_CODE (for table partitioning)

The values for these fields are backfilled during installation of the release, using the values in the parent CCREPREQ record using the CCREPREQREPORT.REP_REQ_ID (FK) as it is never change once created.

Disclaimer: Database efficiency after table partitioning and creation of local indexes on the partitioned tables is very subjective to the size and dispersion of data across laboratory partitions within the CCLAS 6 database, so can vary from server to server.

The application of these partitioned tables and indexes in Datamine's test data environments did not show universal performance improvements, as in some cases performance was improved, but in other, were degraded.

It is advisable for customers to test the partitioning in a non-production environment that mimics the size and dispersion of data across laboratories and load conditions with an actual production environment. It could be that improvements in efficiency could be achieved.

CCLAS-6232 Add ArchiveStatus to database queries within the AppendQC service that impact the Sample Scheme Analyte table

Issue

The AppendQC service needs to run faster.

Investigation

There are some queries to the Sample Scheme Analyte table in the AppendQC service that do not utilise ArchiveStatus. Adding ArchiveStatus to these queries can assist Oracle in selecting an index that results in faster querying.

Outcome

The CCJobService_updateQC sub-services included in the overall AppendQC service on jobs is enhanced by adding the current job's Archive Status to the relevant queries:

- findSampleSchemeAnalytesForSampleScheme
- getLastAnalyteRegAndProfileSequenceForSampleScheme
- updateNumberOfAnalytes

CCLAS-6239 Add ArchiveStatus to database queries within the AppendQC service that impact the Sample Scheme table

Issue

The AppendQC service needs to run faster.

Investigation

There are some queries to the Sample Scheme table in the AppendQC service that do not utilise ArchiveStatus. Adding ArchiveStatus to these queries can assist Oracle in selecting an index that results in faster querying.

Outcome

The CCJobService_updateQC sub-services included in the overall AppendQC service on jobs is enhanced by adding the current job's Archive Status to the relevant queries:

- findByJobSchemeBusinessKey_no118NAndCustomAttributes
- handleOnlyJobDefined
- updateNoOfSchemes
- as.impl.domain.job.count.SampleCountService
- handleOnlyJobDefined
- updateNoOfSchemesStarted
- updateAnalytesRemaining

Batch Job Processing

CCLAS-6213 Implement a Retry mechanism to address problems when there are system batch job failures

Issue

System batch jobs can fail for a number of reasons. The exit messages displayed in the **Steps** tab on the **MSEJEX—Job Executions** screen typically do not show the root cause of the failure. Where a system batch job fails due to a temporary disruption to the network or database connection, then a retry mechanism can assist in ensuring the system batch job runs to completion.

Investigation

A retry that impacts CCLAS tasklet can only be configured in chunk-oriented steps in ventyx-m2000-batch.xml. Retry attempts are not logged in exit messages because each tasklet handles errors differently. However, since every system batch job passes through CCLAS tasklet, then the Retry policy in Spring Batch could be added to this tasklet, for example:

Retry policy applied in CCLASTasklet

```
// Retrieve the retry limit from a system property or default to 3 if not set.
int retryLimit = Integer.parseInt(System.getProperty("cclas.batch.retry.limit", "3"));

RetryTemplate retryTemplate = new RetryTemplate();
SimpleRetryPolicy retryPolicy = new SimpleRetryPolicy();
retryPolicy.setMaxAttempts(retryLimit);
retryTemplate.setRetryPolicy(retryPolicy);

return retryTemplate.execute(context -> {
    if (context.getRetryCount() > 0) {
        log.warn("Batch Execution {} [{}] retry attempt #{} due to exception: {}",
            stepContribution.getStepExecution().getJobExecutionId(),
            stepContribution.getStepExecution().getStepName(), context.getRetryCount
            (), context.getLastThrowable().getMessage());
        return monitoredExecuteInContext(stepContribution, chunkContext);
    }
});
```

Outcome

When a system batch job fails, then a retry mechanism is in place to attempt to get the system batch job to completion.

Retries occur within the same system batch job execution and do not create new system batch job on the system batch queue.

The only impact is that a system batch job may take longer to reach a FAILED status if all retry attempts are exhausted.

This is achieved by implementing the Retry policy in Spring Batch in CCLASTasklet, ensuring that all system batch jobs processed through this tasklet automatically attempt re-execution if they encounter transient failures.

The retry mechanism runs in the Java backend, so users will not notice that a retry was made in the **Steps** tab on the **MSEJEX—Job Executions** screen, however, a warning log entry is made to the system batch job execution log:

System batch job execution log

```
log.warn("Batch Execution {} [{}] - Retry attempt #{} due to exception: {}",
    stepContribution.getStepExecution().getJobExecutionId(),
    stepContribution.getStepExecution().getStepName(),
    context.getRetryCount(),
    context.getLastThrowable().getMessage());
```

This captures the job ID, step name, retry count, and exception message for easy tracing.

The retry limit is set in the **manifests/config/cclas-startup.properties** configuration using the **cclas.batch.retry.limit** system property, for example:

```
cclas.batch.retry.limit=3
```

The default value of **cclas.batch.retry.limit** is 3 if the property is not set.

Note that the logging count is always one less than the value set for **cclas.batch.retry.limit** because the first attempt does not have a previous failure to log, so no retry message is generated on the initial run, and it would be misleading to add 1 to the count as that could trigger unintended retries, especially if **cclas.batch.retry.limit** is set to 0, in which case no retries should occur. That is, this setting represents a retry limit rather than a retry count.

CCLAS-6214 Upgrade Spring Batch 3 to 4

Issue

CCLAS 6 utilises Spring Batch for batch job executions. The CCLAS 6 code should use the latest version of Spring Batch.

Investigation

With the upgrade to Spring Framework 5 and Spring Boot 2, Spring Batch needs to be upgraded from version 3 to version 4.

This gives the advantage in Java configuration, in that, Spring Batch 4 fully supports Java configuration. This means batch jobs can be configured using Java code instead of XML which can make configuration more type-safe and easier to re-factor. This also means we can re-factor batch job configuration from `ventyx-m2000-batch.xml`.

Outcome

CCLAS 6 is updated to use Spring Batch 4. [Spring Batch - Reference Documentation](https://docs.spring.io/spring-batch/docs/4.0.x/reference/html/index.html) (<https://docs.spring.io/spring-batch/docs/4.0.x/reference/html/index.html>).

System Logging

CCLAS-4888 Enhance quality of system logging

Issue

Log messages historically were single line text strings. A common format is

```
(date time) (java class with initials of package) ("level")
(message)
2022-01-01 12:34:56.123 a.d.w.g.ClassName INFO The system has
started
```

These were sent to files on a system, sometimes different files for different aspects - application, server, security.

Trace logs included activity from other users as well as the current user.

With the introduction of cloud servers and aggregators, logs are more often directed to the "console" (SYSOUT) and left to the container or an aggregator watching the container to collect and sort.

Given that the logs are all sent to console, there are no longer separate files to distinguish whether a log message is application, server or security, therefore, that knowledge needs to be included in the log message so that the aggregator can distinguish them.

A simple log example:

```
(date time) (source) (java class with initials of package)
("level") ("correlation-id") (message)
2022-01-01 12:34:56.123 application a.d.w.g.ClassName INFO
12345678-1234-1234-1234-123456789012 The system has started
```

With the addition of more fields, such as prod/int/uat, server node id, etc., these single line text entries become harder to read, and still need predefined parsing configured in the aggregator to provide searchability.

There is a trend towards using JSON format in the logs rather than single lines of text. This makes the logs harder to read en-masse as a human, but easier to read singly and manage via an aggregator, for example:

```
{
timestamp: "2022-01-01 12:34:56.123",
source: "application",
correlation-id: "12345678-1234-1234-1234-123456789012",
classname: "a.d.w.g.ClassName",
level: "INFO",
message: "The system has started"
}
```

We need to investigate suitable additional fields to describe the message categories: application, server, security, etc, and decide on whether plain text suits our customer's needs, or whether CCLAS 6 moves to JSON logging.

Investigation

Structured logs and log aggregators were investigated. CCLAS 6 uses Java's deeply featured logging based on the logback standard library which is an implementation of the slf4j - Simple Logging Facade for Java. CCLAS 6 employs various constructs in this library to provide rich logging to the server manager.

Logs are spat at sys out at the docker's container's boundary, and the aggregator is responsible for reading all of the docker containers. Determine the additional fields so that the log aggregator can distinguish where the log came from.

The logging call does not need to change, it is the definition of the logging format that needs to change, and to have the ability to switch logging to txt or json.

The logging xml file defines the fields included with the log, e.g., 2022-01-01 12:34:56.123 a.d.w.g.ClassName

Outcome

Log message formats suitable for aggregators are defined, a basic log aggregator can be set up, and the UI Debug Logger only reports the current user's run session. Refer to Configuring System Logging and Producing a Session Log in the Online Help.

Scripting

CCLAS-5429 Upgrade scripting to Groovy Version 3

Issue

CCLAS 6 should use the latest version of Groovy.

Investigation

Spring Framework 5 requires Groovy 3, Groovy 3 requires Java 11, and Java 11 requires Spring Framework 5. So these three libraries must be upgraded together.

This upgrade applies to core code, script customisations, functional tests and performance tests.

Note: As at 27th March 2024, the latest version of Groovy was 3.0.22, however this was released after this CCLAS 6 issue was done.

Refer to:

- [The Apache Groovy programming language - Groovy 3.0 release notes \(groovy-lang.org\)](https://groovy-lang.org/release-notes)
- [The Apache Groovy programming language - Download](#)

Outcome

CCLAS 6 is upgraded to be compatible with version 3 of the Apache Groovy programming language. This version of Groovy should be used for all CCLAS 6 Groovy scripts.

Development

CCLAS-4307 Upgrade CCLAS 6 Java to Version 11

Issue

CCLAS 6 should use the latest version of Java.

Investigation

Spring Framework 5 requires Groovy 3, Groovy 3 requires Java 11, and Java 11 requires Spring Framework 5. So these three libraries must be upgraded together.

Note: This is ONLY to be done as a pathway to Java 17. It has no merit for us on its own and in fact brings forward a shorter end-of-life date than Java 8.

Outcome

CCLAS 6 is upgraded to use Java 11.

Refer to [Java JDK Releases](https://www.java.com/releases/) (https://www.java.com/releases/), [Oracle Java SE Support Roadmap](https://www.oracle.com/java/technologies/java-se-support-roadmap.html) (https://www.oracle.com/java/technologies/java-se-support-roadmap.html) and [Amazon Corretto FAQs](https://aws.amazon.com/corretto/faqs/#support) (https://aws.amazon.com/corretto/faqs/#support).

Datamine enables efficient and sustainable mining through the application of world-leading technology and services.

Read the Docs

docs.dataminesoftware.com

Get in Touch

www.dataminesoftware.com/contact

www.dataminesoftware.com/support



Find Us

AUSTRALIA | BRAZIL | CANADA | CHILE | CHINA |
ECUADOR | GHANA | INDIA | INDONESIA | KAZAKHSTAN
| MALAYSIA | MEXICO | MONGOLIA | PERU | PHILIPPINES
| SOUTH AFRICA | TURKEY | UNITED KINGDOM | USA |
UZBEKISTAN



www.dataminesoftware.com