Java API - Release Notes

Release notes

Version 3.3

December 09, 2019

Legal Notices

.

Contents

Contents

Overview

These Release Notes provide information about new features, enhancements, fixes, and changes introduced in the ION Java API from version 125.

Scope

This document refers to the following versions of the API:

Table 1: Java API versions and descriptions

Version	Summary	Release Date
167p6	Support for user impersonation for On-Behalf-Of registration, fix for JSON logging format, minor fixes	09 November 2021
167p5	Support for user impersonation for On-Behalf-Of registration, support for JSON logging to stdout, fixes	07 September 2021
167p4	Entitlement cache invalidation on a per-user basis, user name in functions and transaction calls, fixes	10 June 2021
167p3	Fix for extended UTF-8 characters in strings, obfuscation of sensitive information in registration on-behalf-of requests, ability to log to stdout	13 April 2021
161p4	For MQ stuck in REPLAY loop	15 December 2020
167p2	Fix for SSL data connection drop in case of large DTBLOCKSIZE configuration	14 December 2020
167p1	Simplified ES User Rule Report, UTF-8 support on System Messages, minor fixes	02 November 2020
167	Automatic deadlock detection, improved external authentication protocol, and several fixes	21 August 2020
152p1	Retrieval of User/Group associations based on Pub/Sub protocol	06 April 2020
166	Enhanced setting property management and API for onbehalf-of registration, fixes on Message Queues	02 April 2020
165	Delegation of recordset subscription handling to Router, report of publication object collisions to SysAdmin, and several fixes	06 February 2019
164p1	Fix for NullPointerException on API shutdown, fix for MQ publisher data replay	09 December 2019
163p4	Removed extra dependency from the generated jar	28 October 2019

164	Retrieval of User/Group associations based on Pub/Sub protocol, API for secure download of settings from the Password Server	03 September 2019
163p2	Fixed race condition in multiple MQ subscription pause ()/resume ()	13 August 2019
163p1	GC-friendly management of publications, fixed MQ subscription resume, solved race condition in ES Library initialization, minor other fixes	24 July 2019
163	Entitlement namespace status notification, entitlement cache invalidation with namespace granularity	21 May 2019
162	SSL encrypted connection with Router_Ms, End-To-End statistics for snapshots, Tinfo traces in every log split, various fixes	01 April 2019
161p1	Fixed for persistent chain subscription when using nonexisting field names	07 February 2019
161	Optimized protocol for function calls and sparse supply, support for extended caller information, fix for partial updates on MessageQueue, minor fixes	23 November 2018
152	Support for Daemon Log Archiver, adaptive asynchronous logging, fix for spurious Chain IDLE, minor fixes	27 July 2018
151	Authentication On-Behalf-Of via One-Time-Password, Notification of Password Policy errors, fixes	15 February 2018
150p7	Fix for chain subscription in case the publication of the MkvType changes	11 October 2017
150p6	Fix for starvation when multiple records are unpublished from concurrent threads, fix for race condition in connection cleanup	21 August 2017

Version	Summary	Release Date
150p2	Fix for potential missing supply, fix for supply notification after unpublish event	26 May 2017
150p1	Statistics traces into all open log files and fix on Message Queues	17 March 2017
150	Recordset support, log timestamps with microsecond precision, memory usage stats after major GC event, ultralow latency message passing for same-host connections	02 December 2016
138p3	Support for NaN supplies	23 August 2016
138p2	Backward compatibility for date and time parameters in function call.	27 June 2016

Version	Summary	Release Date
138p1	Fix for calling ION Bus functions using the required DATE and TIME parameters	16 December 2015
138	Asynchronous supply notifications and fix for calling ION Bus functions with mandatory date and time parameters.	09 December 2015
137p3	ES library fixes: • Profile rule conflicts resolved with DENY permission. • Fix for potential exception during configuration download.	01 December 2015
137p2	• Porting of Java API 137 fixes • Non-Daemon threads were not stopped at shutdown.	23 October 2015
137p1	• Improved Java API performance when a custom thread modifies system properties during the startup phase. • Fix on stop.	28 September 2015
137	Entitlements Server library for local in-memory entitlement rule evaluations.	29 April 2015
136p2	• Change in types handling with duplicate field names. • Fix for potential crash at startup.	24 February 2015
136p1	• Using Daemon Master time for custom statistics. • A new method to report custom end-to-end delays based on supply.	28 January 2015
136	• Custom Stats for end-to-end delays. • MkvTickData API. • Improved memory performance of logging.	19 December 2014
135	Support for generic auditing and fixes	10 September 2014
133p7	• MkvDate utilities. • Fix to prevent registration on two different Daemons.	18 June 2014
134p2	• New accessor for the user type. • New setting to disable logging.	07 February 2014
133p6	Fix to prevent exceptions in the MkvSub implementation.	20 September 2013
134	• UTF-8 encoding support for field values. • Chain indexing. • Various fixes.	14 June 2013
133p5	Fix for exceptions raised when using the MkvSub interface.	11 June 2013
133p4	Fix for deadlock when subscribing local MkvChain objects using the MkvSub interface.	15 May 2013

133	Support for a negotiated shutdown protocol and various enhancements on message queues.	25 October 2012
132	New data recording functionality.	10 July 2012
131p2	Extended Garbage Collection (GC) stats, support for user.init credentials, and fixes	21 April 2012
131p1	Extended and revised Custom Statistics examples	30 January 2012
131	Support for secure registration connection	30 September 2011
129p2	Fix for missing queue updates and other fixes	13 September 2011
130	Generic By-Feature Entitlements and record image persistence	26 May 2011
129p1	Missing queue updates and other fixes	03 March 2011
128p5	Missing queue updates and other fixes	25 February 2011
129	• Enhancements to Message Queues • Support for Entitlements Server 103 • Other improvements and fixes	12 October 2010
128p4	• Persistent chain subscription improvements • Enhanced tracing and statistics • Deadlock detection and reporting	11 October 2010
128p3	• Fix for objects not published between a start and stop event • Other minor fixes and changes • Performance improvements	14 May 2010
128	Support for Data Entitlements and other major enhancements	31 August 2009
127	Support for Message Queues	24 April 2009
125p7	Fix for Memory Leaks	31 October 2008
125p6	Fix for NullPointer Exception	09 September 2009
125p5	Fix on Persistent Subscription	24 July 2007
125p4	Fix on External Authentication with Daemon	03 April 2008
125p1	Changes in the MkvDB Files Saving Policy and Enable Tracing of MkvDB Errors	31 July 2007
125	Support for external authentication	20 June 2007

Component details

- Component Name: api_java
- Supported OS: Java
- Executable File Name: jmkv<version>.jar

Requirements

The Java API supports the Java Development Kit (JDK) Versions 1.6, 1.7, and 1.8.

- For JDK 1.7, the recommended update is JDK 1.7.0_75.
- For JDK 1.8, the recommended update is JDK 1.8.0_101.

Oracle no longer supports JDK 1.6. Therefore, ION does not recommend using this version.

If Java version 1.6 is required for legacy compatibility issues, you should use JDK 1.6.0_45 or a later update.

Since Java API 134, components can report Garbage Collection (GC) duration in addition to other GC statistics. To do this, ensure the `tools.jar` Java package distributed with Server JRE or JDK is loaded on the Java Virtual Machine (JVM) where the component runs.

The support for microsecond delays in end-to-end (E2E) and application round-trips generated by components under UNIX machines was introduced in Java API 136. This feature requires the `.so` library files distributed with the Java API package. The `.so` library files must be put in a folder listed under the `LD_LIBRARY_PATH` environment variable of the Daemon application.

Notes

More detailed requirements are available in the dedicated section of the specific version paragraph since Version 165.

Compatibility information

Error notification for all calls pending response for unpublished function objects

Starting from version 135, the Java API issues an error for each pending function call related to an unpublished function object. However, this is a potential breaking change that could expose bugs in function result handlers that do not handle errors correctly.

In fact, many applications publish functions that do not leverage the function error mechanism. Instead, these applications issue application level errors using special values for regular function results, for example, by returning a string such as `1:error`. This approach is still valid for Java API versions 135 and higher.

Before version 135, client components invoking these functions never received a function error. That is, the `onError` method for their `MkvFunctionCallListener` interface implementation was never invoked.

Because errors have never been issued, these client applications appeared to work correctly even when they did not implement error handling correctly.

As explained above, this is no longer the case and client applications should always implement correct handling of function calls errors.

ION recommends that you review the application code to ensure that function call errors are handled correctly.

Using Daemon Master time for custom stats

Starting with version 136p1, the Java API uses the Daemon Master time as the default timestamp when sending custom statistics information. Previously, the Java API used the application local time.

The following code now reports time in the Daemon Master time zone:

```
MkvCustomStatsTable customStatsTable = ...;
customStatsTable.send(statsSupply);
```

Use the following extended method to force the report time to be in application local time:

```
customStatsTable.send(statsSupply, timestamp);
```

Prevent creation of MkvType with duplicate fields

Starting from Java API version 136p2, when a type is created with duplicated fields, the related constructor now fails with an exception.

When upgrading from an older version of the Java API, you must review application code to ensure that the stricter error checking does not prevent business logic types from being published.

Breaking change on function call invocation

Since Java API 138, components invoking function call with `DATE` and `TIME` parameters should use the following:

```
MkvSupply mkvSupply = MkvSupplyFactory.create( new Object[] {
..., MkvValue.createAsDate(20151127), } );
```

Or:

```
MkvSupply mkvSupply = MkvSupplyFactory.create( new Object[] {
..., MkvValue.createAsTime(183023), } );
```

And the resulting supply contains the `DATE` (or `TIME`) parameter, while in previous versions the corresponding type would have wrongly been INT. As result, this is a potentially breaking change in case the Publisher component explicitly tests the type of the parameter expecting an INT type.

Subscriber component code should be reviewed for consistency with their corresponding Publishers.

Registration to the platform is non-encrypted by default when the component runs on recent JVM versions

Java 8 Update 31

Oracle introduced a breaking change since Java 8 Update 31, for which the minimum value of the Diffie-Hellman key-size is set to 768 bits.

To comply with this security constraint, the registration Daemon shall be upgraded to version 114p1 or higher.

With older versions of the registration Daemon, the ION Java API is not able to leverage SSL for the registration connection and it will automatically fallback to a nonencrypted channel.

This does not have a functional impact on the component apart from introducing a delay of few seconds on the registration process.

However, to continue using SSL with a 128-bit key-size, the `java.security` file of the JDK/JRE must be edited to remove this restriction by changing a specific line.

You must change the following line from:

```
jdk.tls.disabledAlgorithms=SSLv3, DH keySize
```

`< 768` To:

```
jdk.tls.disabledAlgorithms=SSLv3
```

Traces included into all open log files

Starting from version 150p1, the Java API is able to append the following traces into all open log files, including PSH and application log files:

- Pinfo: periodic component information, reporting process statistics and basic performance indicators, traced every STATIME
- Cinfo: periodic connection information, reporting data traffic and basic data volumes indicators, traced every STATIME for each open connection
- HWINFO: periodic environment information, reporting basic hardware information, traced every STATIME
- PropertyName / VarName: Java variables and Java API settings, traced at the beginning of every file and at date changes.

Java 8 Update 201

Oracle introduced a breaking change since Java 8 Update 201, for which the anonymous Diffie-Hellman algorithm is deprecated, no matter the length of the key.

As a result, the Java components running under Java 8 Update 201 will register in clear to their Daemon.

To overcome the problem, the `java.security` file of the JDK/JRE can be modified to remove this restriction changing the following line from:

```
jdk.tls.disabledAlgorithms=SSLv3,      RC4,      DES,
MD5withRSA, DH keySize < 1024, \

    EC keySize < 224, 3DES_EDE_CBC, anon, NULL
```

To

```
jdk.tls.disabledAlgorithms=SSLv3,      RC4,      DES,
MD5withRSA, DH keySize < 1024, \

    EC keySize < 224, 3DES_EDE_CBC
```

Known limitations

The current Message Queue implementation is not re-entrant. Therefore, users should not subscribe queues locally.

Additional information

For more information about the Java API, refer to the following documentation on the ION Tracker website: • *Entitlements Server: User Guide*

- *ION Platform Java API Tutorial*
- *Java API Reference Guide*

  From Version 127, this is included in the Java API release package.
- *Java API Samples*
- *Programming Guide*

Java API release notes

Version 167p6: 09 November 2021

Summary

Support for user impersonation for On-Behalf-Of registration, fix for JSON logging format, minor fixes

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with Data Player version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- Audit Server version 200 or later is required if the generic audit service is provided by the Audit Server component.

- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.

The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router 110 or higher.

SSL encrypted connection with Router requires:

- Daemon version 116 or higher
- Router version 110 or higher

ION Server-Side certificate requires Daemon 116p1 or higher and Router version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Entitlement cache invalidation on a per-user basis requires Entitlement Server version 212 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility

from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router using ION Server-Side certificate requires a Java Key Store file in the Router working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities

from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
> openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export -out ion.srv.p12
> keytool -importkeystore -srckeystore ion.srv.p12 -destkeystore ion.srv.jks -srcstoretype pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Fixes

Fix for JSON logging format

The JSON trace has been fixed for the "timestamp" indicator, the "ThreadId" number and suppressing the timestamp header in the message string.

Unable to process trades on ION 167p5 (Bug id.: FICS-110866)

The ION Java API could fail in subscribing a chain via MkvSubscribeManager::persistentSubscribe, in case of a race condition during the initialization of the Data Entitlement state machine.

Version 167p5: 07 September 2021

Summary

Support for user impersonation for On-Behalf-Of registration, support for JSON logging to stdout, fixes

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with Data Player version 300p4 or later.

Generic audit support entails different requirements depending on configuration:
- Audit Server version 200 or later is required if the generic audit service is provided by the Audit Server component.
- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router 110 or higher.

SSL encrypted connection with Router requires:

- Daemon version 116 or higher
- Router version 110 or higher

ION Server-Side certificate requires Daemon 116p1 or higher and Router version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Entitlement cache invalidation on a per-user basis requires Entitlement Server version 212 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.
To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
   > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router using ION Server-Side certificate requires a Java Key Store file in the Router working directory.
To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
   > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
   > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export -out ion.srv.p12
   > keytool -importkeystore -srckeystore ion.srv.p12 -destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Fixes

FICS-107787 – Fix for NullPointerException in Chain Subscription when Data Entitlement are enabled

In case of multiple subscriptions when Data Entitlements enabled on the Platform, a Chain Subscription could raise a NullPointerException.

Version 167p4: 10 June 2021

Summary

Entitlement cache invalidation on a per-user basis, user name in functions and transaction calls, fixes

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with Data Player version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- Audit Server version 200 or later is required if the generic audit service is provided by the Audit Server component.
- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router 110 or higher.

SSL encrypted connection with Router requires:
- Daemon version 116 or higher
- Router version 110 or higher

ION Server-Side certificate requires Daemon 116p1 or higher and Router version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Entitlement cache invalidation on a per-user basis requires Entitlement Server version 212 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.
To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utilityfrom a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.
Connections with Router using ION Server-Side certificate requires a Java Key Store file in the Router working directory. To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
> openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export -out ion.srv.p12
> keytool -importkeystore -srckeystore ion.srv.p12 -destkeystore ion.srv.jks -srcstoretype pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router working directory. To reduce the time validity of the generated certificate, tune the days parameter.

Fixes

Fix for Persistent Chain Subscription

When using MkvSubscribeManager::persistentSubscribe on a Chain, the events onUnpublish were not notified to the application if the Publisher disconnected or became unvisible on the Platform.

Version 167p3: 13 April 2021

Summary

Fix for extended UTF-8 characters in strings, obfuscation of sensitive information in registration on-behalf-of requests, ability to log to stdout

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with Data Player version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- Audit Server version 200 or later is required if the generic audit service is provided by the Audit Server component.
- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router 110 or higher.

SSL encrypted connection with Router requires:
- Daemon version 116 or higher
- Router version 110 or higher

ION Server-Side certificate requires Daemon 116p1 or higher and Router version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.
To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.
To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router using ION Server-Side certificate requires a Java Key Store file in the Router working directory.
To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
> openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export -out ion.srv.p12
> keytool -importkeystore -srckeystore ion.srv.p12 -destkeystore ion.srv.jks -srcstoretype pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Fixes

Support for UTF-8 in setting values

Java API was not able to properly read setting values from the mkv.jinit file if they contain multi-byte UTF-8 characters.

Obfuscation of sensitive information in registration on behalf of requests

When sending a registration request with the on-belhalf-of protocol, the ION Java API traced the sensitive user credentials in the log.

Fix for extended UTF-8 characters in strings

The ION Java API could cause malformed messages on the ION Platform when serializing strings with multi-byte UTF-8 characters, which representation consist of more than 3 bytes (a.k.a. surrogate pairs).

Version 167p2: 14 December 2020

Summary

Fix for SSL data connection drop in case of large DTBLOCKSIZE

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.
Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with Data Player version 300p4 or later.

Generic audit support entails different requirements depending on configuration:
- Audit Server version 200 or later is required if the generic audit service is provided by the Audit Server component.
- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router 110 or higher.

SSL encrypted connection with Router requires:
- Daemon version 116 or higher
- Router version 110 or higher

ION Server-Side certificate requires Daemon 116p1 or higher and Router version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.
To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.
Connections with Router using ION Server-Side certificate requires a Java Key Store file in the Router working directory.
To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
  > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
  > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export -out ion.srv.p12
  > keytool -importkeystore -srckeystore ion.srv.p12 -destkeystore ion.srv.jks -srcstoretype pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router working directory.
To reduce the time validity of the generated certificate, tune the -days parameter.

Fixes

SSL data connection drop in case of large DTBLOCKSIZE configuration

When configuring the DTBLOCKSIZE setting to allow large packet delivery (more than about 80 KB) and also enabling at the same time the SSL encryption over data connection, the application could unexpectedly cause a socket drop.

Version 167p1: 02 November 2020

Summary

Simplifies ES User Rule Report, UTF-8 support on System Messages, minor fixes

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with dplayer version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- audit_server version 200 or later is required if the generic audit service is provided by the audit_server component.

- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router_M 110 or higher.

SSL encrypted connection with Router_Ms requires:
- Daemon version 116 or higher - Router_M
version 110 or higher.

ION Server-Side certificate requires Daemon 116p1 or higher and Router_M version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
   > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out
ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
   > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
   > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export -out ion.srv.p12
   > keytool -importkeystore -srckeystore ion.srv.p12 -destkeystore ion.srv.jks srcstoretype
pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Enhancements

Simplified ES User Rule Report

Using MkvByFeatureReportQoS with 'simplifiedReport' flag, allows to retrieve a User Entitlement Rule report including all rules (allow/deny and conflicting rules) without any pre-processing.

UTF-8 support on System Messages

The ION Java API now supports UTF-8 multi-byte characters strings in System Messages sent to the Platform via MkvPlatform::sysMessage(int level, String message)

Version 167: 21 August 2020

Summary

Automatic deadlock detection, improved external authentication protocol, and several fixes

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with dplayer version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- audit_server version 200 or later is required if the generic audit service is provided by the audit_server component.

- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router_M 110 or higher.

SSL encrypted connection with Router_Ms
requires: - Daemon version 116 or higher -
Router_M version 110 or higher.

ION Server-Side certificate requires Daemon 116p1 or higher and Router_M version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.
To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.
To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
> openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -x509 -days 3650 -out ion.srv.crt
> openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export -out ion.srv.p12
> keytool -importkeystore -srckeystore ion.srv.p12 -destkeystore ion.srv.jks -srcstoretype pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

New Features and Changes

Support for improved external authentication protocol

The ION Java API supports new methods to request the creation of an authentication session, which is applicable to systems using an external authenticator based on OpenID Connect or OAuth 2.0.
The new API methods are as follows:

```
MkvPlatform::createAuthenticationRequest(
      String replyURL,
      String context,
      String serviceId,
      String extraParameters,
      MkvAuthenticationRequestCreationListener listener
   )

   MkvPlatform::createAuthenticationRequest(
      String replyURL,
      String context,
      String serviceId,
      String compType,
      String host,
      String hwInfo,
      String extraParameters,
      MkvAuthenticationRequestCreationListener listener
   )
```

Automatic deadlock detection

The ION Java API automatically detects an internal thread deadlock condition. If this happens, the current thread status is dumped to log files and the application stops.This prevents higher and wider business impact, in case the application is impaired to continue its execution effectively, due to a thread deadlock condition.

Fixes

Unable to reply to transaction requests with UTF- 8 messages

Now the ION Java API properly supports UTF-8 strings in transaction call results.

Lost very long system messages

Now the ION Java API automatically truncates system messages (send by client applications) to the maximum allowed length of 1024 bytes. This prevents loosing the system messages in case of very long content.

Multiple subscriptions of a message queue

The middleware infrastructure allowed multiple subscribers to subscribe to a message queue available on the bus, causing unexpected behaviors such as multiple notifications of the same records.Now the ION Java API guarantees unique subscriber for any message queue object.

This fix requires the availability of the Router 111p5 (or higher) in the infrastructure.

Version 152p1: 06 April 2020

Summary

Retrieval of User/Group associations based on Pub/Sub protocol

Requirements

ION api_java supports the following Oracle JDK's:
JDK 1.6, JDK 1.7 and JDK 1.8

JDK 1.7.0_75 and JDK 1.8.0_101 are the latest recommended updates for Java version 7 and 8 respectively.
JDK 1.6 is no longer supported by oracle and is not recommended.
Should Java version 6 be needed for legacy compatibility issues, JDK 1.6.0_45 or a later update should be used.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with dplayer version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- audit_server version 200 or later is required if the generic audit service is provided by the audit_server component.

- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Enhancements

Retrieval of User/Group associations based on Pub/Sub protocol

The retrieval of User/Group associations is now based on Pub/Sub protocol, using the required Daemon chains.This overcomes existing limitations for Users belonging to more have ~50 different groups. Thanks to this enhancement, the API

MkvUser::getGroups, MkvUser::getGroupIDs, MkvUser::belongsToGroup

And

MkvGroup::getUsers, MkvGroup::getUserIDs

are guaranteed to return comprehensive results, even in case of high number of users/group distinct associations in the Platform.

Version 166: 02 April 2020

Summary

Enhanced setting property management and API for on-behalf-of registration, fixes on Message Queues

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.

Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with dplayer version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- audit_server version 200 or later is required if the generic audit service is provided by the audit_server component.

- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.
Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router_M 110 or higher.

SSL encrypted connection with Router_Ms requires:
- Daemon version 116 or higher - Router_M
version 110 or higher.

ION Server-Side certificate requires Daemon 116p1 or higher and Router_M version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
 > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the "`ion.srv.crt`" and "`ion.srv.key`" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

    >    keytool    -importkeystore    -srckeystore
ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.
To reduce the time validity of the generated certificate, tune the -days parameter.

New Features and Changes

Ability to override the default value of a setting property

Starting from Java API 166, the application can override the default values of any setting property, even properties registered by other ION libraries, using the MkvQoS interfaces:

```
    MkvQoS::overridePropertyDefaultValue(
String settingName,         int newDefault)
    MkvQoS::overridePropertyDefaultValue(
      String settingName,
      String newDefault)
```

If the property is not explicitly configured, it takes the new overridden default value, independently of the default value specified at registration time (for example, with MkvProperties::registerProperty()).

Ability to link the values of a property to another property

Starting from Java API 166, the application can link the values of any property to the value of another property, using the MkvQoS interface:

```
    MkvQoS::linkPropertyValues(
      String settingName,
      String referencedSettingName)
```

If the property is not explicitly configured, it takes the same value as "referencedSettingName", independently of the default value specified at registration time (for example, with MkvProperties::registerProperty())

Optional QoS for by-Feature Entitlement Report API

A new API is available to produce an in-memory By-Feature Entitlement Report:

```
MkvEntitlementManager::generateByFeatureReport(
   String entitlementNamespace,
   String userName,
   String feature,
   MkvByFeatureReportQoS qos,
   IMkvByFeatureEntitlementReportListener listene)
```

The "MkvByFeatureReportQoS" parameter can be used to drive the behavior of the report. If null or default, the report will automatically execute a post-processing, suppressing redundant/inheritance rules.

If passed with the new flag "simplifiedReport" to true, as in:

```
new MkvByFeatureReportQoS(true)
```

the report will contain all raw rules, without an filter or post-processing.

Registration on-behalf-of API

Starting from Java API 166, applications can register virtual components on the platform through the new on-behalf-of registration API.

The following interfaces are available, to register a virtual component using classic user credentials:

```
MkvPlatform::registerOnBehalfOfUsingPassword(
   String username,
   String password,
   String newPassword,
   String otp,
   String compType,
   String host,
   String hwInfo,
   MkvOnBehalfOfAuthenticationWithPasswordResponseListener authenticationListener)

MkvPlatform::registerOnBehalfOfUsingPassword(
   String username,
   String password,
   String newPassword,
   String otp,
   String compType,
   String host,
    MkvOnBehalfOfAuthenticationWithPasswordResponseListener authenticationListener)
```

The following interfaces are available, to register a virtual component using enterprise authentication token:

```
MkvPlatform::registerOnBehalfOfWithAuthenticationToken(
  String serviceId,
  String tokenEncoding,
  String tokenValue,
  String otp,
  String compType,
  String host,
  String hwInfo,
  MkvOnBehalfOfAuthenticationWithTokenResponseListener authenticationListener)

MkvPlatform::registerOnBehalfOfWithAuthenticationToken(
  String serviceId,
  String tokenEncoding,
  String tokenValue,
  String otp,
  String compType,
  String host,
  MkvOnBehalfOfAuthenticationWithTokenResponseListener authenticationListener)
```

This feature requires the Daemon version 115p3 (or higher).

Fixes

java.io.EOFException in jmkv164p1.jar (Issue id.: FICS-89481)

On application shutdown, an EOF Exception could be thrown with a stack similar to the following:

```
11:32:32:764 CET 2020 java.io.EOFException
11:32:32:764 CET 2020 at com.iontrading.a.a.c(ASocketChannelDTBlockWriter.java:446)
11:32:32:764 CET 2020 at com.iontrading.mkv.ck.a(NetOutputStream.java:121)
11:32:32:764 CET 2020 at com.iontrading.mkv.ck.a(NetOutputStream.java:336)
11:32:32:764 CET 2020 at com.iontrading.mkv.ai$e.a(MkvConnection.java:5355)
```

The exception had no impact on business functionality, but appeared in the component logs.

Version 165: 06 February 2020

Summary

Delegation of recordset subscription handling to Router, report of publication object collisions to SysAdmin, and several fixes

Requirements

The ION Java API requires a Java Runtime to run and it has been tested using the following vendors and versions of the Java Development Kit (JDK):

- Oracle JDK 8
- Oracle JDK 11 (recommended)
- Amazon Corretto 8
- Amazon Corretto 11

Although there should be no issue with using other JDKs that are fully compatible with the Java specifications (e.g., Red Hat OpenJDK), ION cannot guarantee the correct behavior of the component using a vendor or a version other than those mentioned above. ION is committed to keeping this list of tested JDK updated based on the Oracle's roadmap and the actual market trends.
Generic Data Entitlements support requires Entitlement Server 105, Daemon 105, CS 133.

Secure registration channel requires Daemon 106p4 or later.

Shutdown protocol support requires Daemon 108 or later.

The format of data recordings is compatible with dplayer version 300p4 or later.

Generic audit support entails different requirements depending on configuration:

- audit_server version 200 or later is required if the generic audit service is provided by the audit_server component.

- Daemon version 110 or later is required if the generic audit service is provided by the Daemon infrastructure. In contrast, dedicated audit components entail no special requirement on the Daemon infrastructure.

The support to microseconds delays in End-To-End and Application Round-Trip generated by components under UNIX machines requires the .so library files distributed along with the Java API package.
The .so library files should be put in a folder listed under the LD_LIBRARY_PATH environment variable of the Daemon application.

The support to on-behalf-of by-feature license API requires Daemon 114 or higher.

Ultra-low latency message passing for same-host connections requires Java 8 or higher.

Unique identification of ION Platforms requires Daemon 114p1 or higher.

Entitlement Server library requires Entitlement Server version 108 or higher.

Caller info on function and transaction requests require Router_M 110 or higher.

SSL encrypted connection with Router_Ms requires:
- Daemon version 116 or higher - Router_M
version 110 or higher.

ION Server-Side certificate requires Daemon 116p1 or higher and Router_M version 110 or higher.

Entitlement namespace status notification requires Entitlement Server version 202 or higher.

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
 > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

    >    keytool    -importkeystore    -srckeystore
ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Enhancements

Delegation of recordset subscription handling to Router

Starting from version 165, recordset publishers can delegate the Router component to automatically reply to all subscription requests via recordset, providing the current local data snapshot for the requested fields if available.

If the data snapshot is already available in the Router, the Router will deliver it to the subscriber without interacting with the publisher any more.

This behavior allows the publisher to delegate the overhead of subscription handling via recordset to the Router, like for regular records.

To leverage the feature, the recordset publisher needs to explicitly provide the Router permission to accept subscriptions. This can done using the following Java API's class constructors:

MkvRecordset(String name, String type, boolean allowRouterToAcceptAllSubscriptions)

MkvRecordset(String name, String type, boolean allowRouterToAcceptAllSubscriptions, MkvSubscriptionRequestListener listener)

Report of publication object collision to SysAdmin

Starting from version 165, if a Java API application receives a publication object twice, due to a misconfiguration of the platform or a topological problem (i.e., duplicated path), the API automatically reports an error notification message to the System Administrator Tool (SysAdmin), visible on the Platform Event Viewer panel.

The message is of one of the following form: Object collisions found

followed by one of the following reasons which triggered the problem:

- platform topology should be reviewed to remove duplicated paths
- components configuration should be checked to remove source clashes
- components configuration should be checked to remove conflicting objects
- components configuration and platform topology should be reviewed

The message also includes a detailed report about the affected originator(s) and connection(s) which contributed to the duplicated publication.

Those messages can also be intercepted by the Performance Meter Tool (PerfMeter) to trigger an alert to administrator personnel.

New Features and Changes

Deprecation of old API for on-behalf-of registration

The old API interfaces for on-behalf-of registration

(MkvPlatform::registerOnBehalfOf) has been deprecated and it will disappear in a next version.

Please contact the ION Support for further information if you require this functionality.

Fixes

Missing data on MQ subscriber using pause-resume feature

When a Subscriber resumes a MQ, the Publisher is forced to perform a REPLAY of old data, to re-align the Subscriber.

If at the same time the publisher application starts pushing new data into the MQ, the protocol could get stuck resulting in missing data on the Subscriber side.

NullPointerException on API shutdown

If the API is shutdown during a publication download (e.g., while serving a connection from a Router), it was possible that few NullPointerExceptions were logged into the PSH logs due to a race condition with the API shutdown.

Deadlock in log traces

When two different threads tried to produce traces while the local machine Timezone was modified, the Java API could generate a deadlock.

PXE missing subscription of Reuters SFC records (Issue id.: FICS86856)

A NullPointer exception was thrown by the Java API when:

- a subscriber application performed a MkvSubscribeManager::persistentSubscribe() for a record
- the record was published but the type was not yet available - the record was unpublished right away

Version 164p1: 09 December 2019

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
 > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

     > keytool      -importkeystore      -srckeystore

ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Summary

Fix for NullPointerException on API shutdown, fix for MQ publisher data replay

Fixes

Fix for NullPointerException on API shutdown

With the previous versions of the ION JAVA API, if the API is shutdown during a publication download (e.g.: while serving a connection from a Router), it is possible that a few NullPointerExceptions are logged into the PSH logs due to a race condition with the API shutdown.
Fixed.

Fix for MQ publisher data replay

An MQ publisher can re-send old data to the Subscriber if required (e.g.: the Subscriber did not process data in-order).

If, at the same time, the application produces new data in the MQ, the publisher sends out scramble acknowledgement requests to the Subscriber, resulting in un-alignment or loss of data on the Subscriber side.

Fixed.

Version 163p4: 28 October 2019

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
  > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the `"ion.srv.crt"` and `"ion.srv.key"` files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl  req  -newkey  rsa:2048  -nodes  -keyout  ion.srv.key  -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

    >    keytool     -importkeystore     -srckeystore
ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the `"ion.srv.jks"` file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Summary

Removed extra dependency from the generated jar

Fixes

Removed extra dependency from the generated jar

The final ION Java API jar contained extra dependency files which were not required. Fixed.

Version 164: 03 September 2019

Summary

Retrieval of User/Group associations based on Pub/Sub protocol, API for secure download of settings from the Password Server

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
 > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

    >    keytool    -importkeystore    -srckeystore
ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Enhancements

Retrieval of User/Group associations based on Pub/Sub protocol

The retrieval of User/Group associations is now based on Pub/Sub protocol, using the required Daemon chains. This enhancement overcomes existing limitations for users belonging to more than ~50 different groups. Thanks to this enhancement, the API MkvUser::getGroups, MkvUser::getGroupIDs, MkvUser::belongsToGroup and

MkvGroup::getUsers, MkvGroup::getUserIDs

are guaranteed to return comprehensive results, even in case of high number of users/group distinct associations in the Platform.

New Features and Changes

API for secure download of settings from the Password Server

MkvProperties exposes a new API for secure downloading of settings from the Password Server:

```
                MkvProperties::getServicePassword(String
serviceName, MkvServicePasswordListener listener)
```

The API is meant to be used to remove the need for sensible password settings in the `mkv.jinit` file.

Sensible password settings can then be downloaded directly from the Password Server on the secure registration channel with the Daemon.

Fixes

Fix for failure of MQ subscriptions

Subscriptions to MQ failed if the PSH logs were forcibly disabled.

Fix for transactions on Recordset data

With the previous versions of the JAVA API, the publisher application was not notified about transaction requests on records subscribed via Recordset.

Version 163p2: 13 August 2019

Summary

GC-friendly management of publications, fixed MQ subscription resume, solved race condition in ES Library initialization, minor other fixes

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
  > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl  req  -newkey  rsa:2048  -nodes  -keyout  ion.srv.key  -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

    >    keytool     -importkeystore     -srckeystore
ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Fixes

Multiple pause ()/resume () on a MQ subscription could result in missing data

Using pause / resume on a Message Queue subscription might result in missing data. Specifically, the subscriber might never receive the data sent by the publisher after the pause and before the resume, if multiple pause/resume have been invoked within the same MQ flow window.
Fixed.

Version 163p1: 24 July 2019

Summary

GC-friendly management of publications, fixed MQ subscription resume, solved race condition in ES Library initialization, minor other fixes

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
 > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the `"ion.srv.crt"` and `"ion.srv.key"` files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl  req  -newkey  rsa:2048  -nodes  -keyout  ion.srv.key  -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

    >    keytool     -importkeystore     -srckeystore
ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the `"ion.srv.jks"` file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Enhancements

GC-friendly management of record publications

On record publications, the Java API generates temporary strings to determine whether a Recordset object exists with the same object name prefix.

Starting from version 163p1, this process is now more GC-friendly, avoiding the creation of unnecessary temporary strings.

Timestamped trace for "Logging Suspended" and "Logging Resumed"

Starting from version 163p1, the Java API traces the "Logging Suspended" and "Logging Resumed" information with the full timestamp header, similarly to all other traces.

Fixes

Fix for ES Library remaining stuck on ION Entitlement Server bounce

Due to a race condition, when the ION Entitlement Server is bounced, there is chance that the ES publications arrive to the Java API before the ES event register.

In such scenario, a NullPointerException was thrown, resulting in the Enitlement state machine to remain stuck.

Fixed.

MQ subscription pause/resume could result in missing data

Using pause / resume on a Message Queue subscription might result in missing data.

Specifically, the subscriber might never receive the data sent by the publisher after the pause and before the resume, in case the following conditions hold:

- The pause has been invoked right after the flush of the last MQ window.
- The new data produced after the pause does not fill the new window. • The publisher does not flush nor close the MQ.

Fixed.

Version 163: 21 May 2019

Summary

Entitlement namespace status notification, entitlement cache invalidation with namespace granularity

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
 > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

This generates the "ion.srv.crt" and "ion.srv.key" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
    > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -
x509 -days 3650 -out ion.srv.crt

     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12

    >    keytool    -importkeystore    -srckeystore
ion.srv.p12 - destkeystore ion.srv.jks -srcstoretype
pkcs12
```

This generates the "ion.srv.jks" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

New Features and Changes

Notification of availability for By-Feature entitlement namespaces

Starting from version 163, the ION Java API notifies the availability of each By-Feature entitlement namespace on the specific listeners installed via:

```
MkvEntitlementNamespace::addMkvESNamespaceAvailabilityListener(
                IMkvESNamespaceAvailabilityListener listener)
```

The application can then react to the following events:

- onNamespaceActive(IMkvNamespace namespace);
- onNamespaceInactive(IMkvNamespace namespace);

The ION Java API notifies the status of each By-Feature entitlement namespace, including those managed internally by the ION Entitlement Server as well as those externally delegated to a third-party system.

Internal namespaces will become active as soon as the ION Java API has connected to the ION Entitlement Server and it is ready to serve queries on those namespaces.

Externally delegated namespaces will switch to active or inactive based on the external system status.

For details about the delegation to external systems, their status record and further information, please refer to:

https://client-connect.iongroup.com/library/content/core-technology/p9077836/p9085122/p11671578/delegation-to-an-externalsystem/

Entitlement cache invalidation for externally delegated namespaces

Starting from version 163, the Java API reacts to changes in the configuration of any external system used for entitlement delegation.

A change in the external system status triggers the invalidation of the local entitlement cache.

For details about the delegation to external systems, their status record and further information, please refer to:

https://client-connect.iongroup.com/library/content/core-technology/p9077836/p9085122/p11671578/delegation-to-an-externalsystem/

Entitlement cache invalidation on a per-namespace basis

Starting from version 163, the ION Java API maintains an independent entitlement cache for each single namespace. Whenever the API detects a change in the configuration of a specific namespace, only the pertinent cache is invalidated.

This behavior applies also to namespaces that are delegated to an external system.

For details about the delegation to external systems, their status record and further information, please refer to:

https://client-connect.iongroup.com/library/content/core-technology/p9077836/p9085122/p11671578/delegation-to-an-externalsystem/

Custom Stats for Heap and Non-Heap memory usage in JVM

Starting from version 163, the ION Java API reports a new set of Custom Statistics to the ION Platform, called "`JVM Heap Stats`". The statistics report the heap space usage and the non-heap space usage (native memory) information as retrieved from the underlying JVM.

The statistics can be visualized using the ION Performance Meter tool via the System Administrator PerfMeter Add-on.

Version 162: 01 April 2019

Summary

SSL encrypted connection with Router_Ms, End-To-End statistics for snapshots, Tinfo traces in every log split, various fixes

Notes

Registration using ION Server-Side certificate requires a certificate and private key files in the Daemon working directory.

To generate the certificate and private key files as required by the ION Platform, OpenSSL can be used, running the following utility from a command prompt:

```
    > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key -
x509 -days 3650 -out ion.srv.crt
```

This generates the "`ion.srv.crt`" and "`ion.srv.key`" files, valid for 10 years (3650 days), that shall be stored in the Daemon working directory.

To reduce the time validity of the generated certificate, tune the -days parameter.

Connections with Router_M using ION Server-Side certificate requires a Java Key Store file in the Router_M working directory.

To generate the Java Key Store file as required by the ION Platform, OpenSSL can be used, running the following utilities from a command prompt:

```
  > openssl req -newkey rsa:2048 -nodes -keyout ion.srv.key x509
-days 3650 -out ion.srv.crt
```

```
     > openssl pkcs12 -inkey ion.srv.key -in ion.srv.crt -export
-out ion.srv.p12
```

```
    >   keytool   -importkeystore   -srckeystore
 ion.srv.p12   -   destkeystore   ion.srv.jks   -
srcstoretype pkcs12
```

This generates the "`ion.srv.jks`" file, valid for 10 years (3650 days), that shall be stored in the Router_M working directory.
To reduce the time validity of the generated certificate, tune the -days parameter.

Enhancements

Report End-To-End statistics for snapshot supplies

End-To-End statistics now include snapshot information.
In this way, even in the absence of incremental updates, the PerfMeter and the Daemon MONITOR logs will report the trace for the Publisher/Subscriber End-To-End.

SSL encrypted connection with Router_Ms

Starting from version 162, the ION Java API supports SSL encrypted, secure connections with Router_M instances.
Encryption uses TLSv1.2 and it requires Router_M version 110 or higher.
To enable encrypted connections with Router_M instances, the following API setting is available:

```
mkv.enablessl=<component list>
```

The component list is a space-separated list of peer components, with which the API will attempt to use SSL handshake.

The wildcard character "*" is allowed, to specify that SSL shall be used on every connection towards any peer component.

Dynamic change of LOGSDAYS configuration

The API can be configured with the maximum number of log days to keep in the logging folder.
The LOGSDAYS variable is available for this customization.
Starting from version 162, the LOGSDAYS variable can be changed at runtime.

New Features and Changes

Report Tinfo traces in every log file split

Timezone information in Tinfo is now reported in every log file split.

Support for ION Server-Side certificate in SSL connections

When establishing an SSL connection for registration, the Java API 162 can support ION Server-Side certificate provided by the Daemon.

ION Server-Side certificate can be used in place of anonymous SSL connections, to verify the identity of the Daemon and avoid Man-in-the-Middle attacks.

Fixes

Support to dynamic setting creation in SysAdmin using MkvPropertyGroup

The application can register a property group via MkvPropertyGroup class.
Starting from Java API 162, the application will get notified about any change of existing properties as well as any new property created in the group, at run-time via SysAdmin.

This allows the application to peak up new settings in the property group without restarting the component.

Fix for ESLib remaining stuck in "Downloading" phase

At the startup, the Entitlement Server library embedded into the Java API could remain stuck in “Downloading" phase, due to a subscription race condition on the Entitlement Server chains. The result would be a failure in all subsequent entitlement queries. Fixed.

Version 161p1: 07 February 2019

Summary

Fix for persistent chain subscription when using non-existing field names

Fixes

Fix for persistent chain subscription when using non-existing field names (Issue id.: FICS-74084)

When using MkvSubscribeManager::persistentSubscribe() on a chain, the application can specify the list of requested field names.
Starting from the Java API version 160, a MkvFieldNotFoundException is thrown if the list contains non-existing field names.
We relaxed this behavior in 161p1: the invalid field names are simply ignored, while the subscription goes through.
Fixed.

Version 161: 23 November 2019

Summary

Optimized protocol for function calls and sparse supply, support for extended caller information, fix for partial updates on MessageQueue, minor fixes

New features and changes

Optimized protocol for function calls

Java API 161 introduces a more efficient, lightweight message for function call requests. The message is serialized into a compressed packet, representing the request from the caller towards the Publisher.

The new protocol is transparent to the applications and results in both bandwidth optimization and better performance for function call latency.

Optimized protocol for sparse supplies

Java API 161 introduces a more efficient serialization for sparse supplies, that is: supplies with a small number of changed fields with respect to the defined record type. The serialization uses a compressed, efficient bitmask to represent the changed fields information.

The new protocol is transparent to the applications and results in better performance for the End-To-End delays.

Extended maximum value for 'logsbuffersize' setting

The 'logsbuffersize' setting controls the size of the in-memory buffer associated to each log file. The maximum configurable value for 'logsbuffersize' is now 500Mbytes.

Caller info in Function and Transaction calls

Starting from version 161, the Java API supports the extended information about function and transaction callers. The information is wrapped into the MkvCallerInfo object and includes: the original caller name, it's authentication token and its InetAddress/IP.

All information is optional and provided on the MkvFunctionCallEvent / MkvTransactionCallEvent only if the caller provided it. The MkvCallerInfo object can be retrieved using:

- MkvFunctionCallEvent::getCallerInfo() for function calls.
- MkvTransactionCallEvent::getCallerInfo() for transaction calls.

Components in between can forward the incoming MkvCallerInfo on any outbound call using the proper overload methods:

- MkvFunction::call().
- MkvRecord::transaction().

Fixes

Fix for the adaptive End-To-End Delay statistics

End-To-End delays are computed by each Subscriber for each received update, to report the overall latency of supply messages, from the Publisher down to the Subscriber. Since Publisher and Subscriber could run on different boxes, the End-To-End are computed based on an adaptive algorithm, that takes into account possible clock skews, drifts and un- alignments.

In case of clock skews, the Subscriber could wrongly report misleading, artifact EndTo-End. Fixed.

Partial updates on MessageQueue

Java API 160 introduced the support for partial updates on MessageQueue. The initial implementation contained a breaking change for older MessageQueue Subscribers, which could potentially lead to ArrayIndexOutOfBounds exception. Fixed.

Version 152: 27 July 2018

Summary

Support for Daemon Log Archiver, adaptive asynchronous logging, fix for spurious Chain IDLE, minor fixes

Enhancements

Adaptive asynchronous logging

Normally, any call to log a new trace via Java API MkvLog is non-blocking: the new trace will be appended into a memory buffer and the control is immediately returned to the application.

A flusher thread asynchronously writes data onto the disk file, without interfering with the application business logic and/or performance.

If the memory buffer is full, the new trace is discarded and the log contains the following trace: "Logging Suspended (Logging buffer full)".

For MiFID II compliance, Java API 152 introduces an adaptive logging strategy. In case the log memory buffer is temporarily full, the logging call becomes blocking: the application will wait for the flusher thread to write the data onto the disk and to make enough room in the buffer to host the new trace.

The application temporarily blocks only if the I/O throughput is reasonable high to cope with the logging pressure. The temporary block is anyway limited by a (configurable) maximum timeout, not to impact the application performance in a sensible way.

In this way, as long as the disk is able to cope with the amount of application log traces, the API can guarantee that no trace will get lost.

See MkvLog Javadoc for further details.

MkvVersion class is kept in clear

The MkvVersion class appears now in clear in the ION Java API jar.

This class contains the build number and date for the ION Java API release.

Support for compressed log files when splitting

When the log file size reaches the configured size limit, the ION API automatically renames it using the next sequential number (i.e.: ".1", ".2", etc.). To compute the next sequential number, the ION API searches for existing files in the logs folder.

Starting from version 152, the ION Java API also takes into account any compressed file, with any of the following extensions: ".gzip", ".gz", ".tar", ".tgz", ".zip", ".7z", ".rar", ".xg".

This integrates seamlessly with the ION Log Archiver utility.

Custom E2E Sample supports virtual destination component

When reporting Custom E2E latency with a peer component, the application can leverage the MkvPlatform::createE2EDelaySampler API and the MkvE2EDelaySampler utility class.

Those API allow the application to specify the originator component only ('From' field):

by default, the destination component of the E2E ('To' field) is the invoking component itself.

Applications can now use the overloaded MkvPlatform::createE2EDelaySampler and MkvPlatform::getE2EDelaySampler API to customize the destination component ('To' field) as well.

New features and changes

Enhanced logging for malformed supply

When invoking a function using a malformed supply, the Java API throws an MkvInvalidSupplyException.

Starting from Java API 152, this exception has become more specific, including the detailed information about which supply field is incompatible with the type.

This will ease the troubleshooting and debugging of ION Java Subscriber applications.

MkvFunctionCallEvent::getRemote deprecated

This method is deprecated since its value is always empty and it is not trasmitted over the protocol.

There is no alternate method to substitute this.

Default Debug level now set to 100

Fixes

Fix for MkvFunction unpublish

When a function gets unpublished and the Subscriber has a pending function call yet to be returned, the Java API notifies a function error with the message:

-1:received unpublish event before function call return

If the Subscriber reacts to the error and uses the MkvAvailabilityService to detect when the function will be available again, the MkvAvailabilityService is triggered immediately, in stack, by mistake.

Fixed.

Spurious chain IDLE event notified in case of frequent Subscription/Unsubscription requests

If the application frequently requests a Chain Subscription following by the corresponding Unsubscribe, a spurious chain IDLE event can be notified to the next Chain Subscription.

Fixed.

Version 151: 15 February 2018

Summary

Authentication On-Behalf-Of via One-Time-Password, Notification of Password Policy errors, fixes

Enhancements

Performance improvements for memory consumption and GC events

Java API 151 implements a few memory usage optimization that are able to drastically reduce the amount of Young Generation garbage in scenarios of high frequency updates.

These improvements result in fewer cycles of Minor GC events at runtime.

Performance improvements for MkvUnicodeBuffer

MkvUnicodeBuffer has been optimized to leverage zero GC recycling of the internal buffer.

Review of Java API samples and packaging

The collection of tutorial and sample applications has been revised, with a standalone script for building and running the samples on any supported O.S.

Refer to the readme.txt file in the sample package for additional details about using the distributed codes.

New features and changes

Switch off the Host IP -> Host Name resolution

By default, the resolution of Host IP to Host Name is now switched off.

Optionally, this feature can be enabled via the following configuration setting:

```
mkv.resolve_hostnames = 1
```

On execution environment where the DNS is not properly configured to perform reverse name lookups, the Host IP -> Host Name resolution could cause disruption in application connectivity and latency in establishing new connections.

Fixes

Race condition on connection cleanup

A race condition could happened in cleaning up the connections towards a remote router component.

The race condition resulted in a NullPointerException to be triggered during the connection cleanup, causing the connection to remain dangling.

Any subsequent connection attempt made from the same remote component (i.e.: after its bounce), could also result in the connection to be refused with an "Already Logged" error.

Fixed.

Version 150p7: 10 November 2017

Summary

Fix for chain subscription in case the publication of the MkvType changes

Enhancements New features and changes Fixes

ION Java API: onFullUpdate() missing fields in supply

In case of disconnection from the Publisher and/or MkvType published again with different layout, a Chain Subscriber could resume the subscription with wrong field ids.

This is due to internal caching of field id masks, which are not re-initialized on MkvType unpublish/publish event.

The effect is the Chain Subscriber could receive fields it didn't need and miss the fields it actual required.

Fixed.

Version 150p6: 21 August 2017

Summary

Fix for starvation when multiple records are unpublished from concurrent threads, fix for race condition in connection cleanup

Enhancements

Performance improvements for memory consumption and GC events

Java API 150p6 implements a few memory usage optimization that are able to drastically reduce the amount of Young Generation garbage in scenarios of high frequency updates.

These improvements result in fewer cycles of Minor GC events at runtime.

New features and changes Fixes

Fix for starvation when multiple records are unpublished from concurrent threads

In case the application performs unpublish on different records from multiple concurrent threads, some of those threads could hang forever, causing starvation.

This was a regression of Java API 150p2. Fixed.

Custom component did not receive updates from router after connection reset event

Fixed.

Version 150p2: 26 May 2017

Summary

Fix for potential missing supply, fix for supply notification after unpublish event

Enhancements

Performance improvements for ESLib

The Java API includes a layer for entitlement queries.

This layer has been modified to provide even faster, higher performance evaluations of Data Entitlement and By-Feature Entitlement requests.

New features and changes

Java API traces entire JVM command line arguments

Starting from version 150p2, the Java API traces the entire JVM command line arguments in its logs, for troubleshooting purposes.

High resolution timestamps on Windows Server 2012 boxes is enabled by default

Starting from Java API 150p2, the default behavior for timestamps on Windows Server 2012 boxes is to rely on the Windows Native API for high resolution timestamps.

The user does not have to configure the explicit setting mkv.e2enative.enable any longer.

Fixes

Fix compatibility issue with subscriber components using C-API version lower than 128 and Java API version lower than 128

API 150 had an incompatibility with subscriber components built on top of C-API versions lower than 128 and Java API versions lower than 128.

The incompatibility was causing supply messages to be discarded on the subscriber side, resulting in missing data.

Version 150p1: 17 March 2017

Summary

Statistics traces into all open log files and fix on Message Queues

Enhancements New features and changes

Unique identification of ION Platforms

The ION Platforms are now uniquely identified improving the reporting features provided by the ION Online Diagnostic and the ION Performance Meter tools.

The unique identifiers are automatically generated by ION Tracker, and they are delivered to the ION Platforms through the new license files.

Components using Java API 150p1 (or higher) report the unique client and platform identifies in their PSH logs (Pinfo traces).

Broadcasting of statistics traces into all open log files

Starting from version 150p1, the Java API is able to append the following traces into all open log files, including PSH and application log files:

- Pinfo: periodic component information, reporting process statistics and basic performance indicators, traced every STATIME

- Cinfo: periodic connection information, reporting data traffic and basic data volumes indicators, traced every STATIME for each open connection

- HWINFO: periodic environment information, reporting basic hardware information, traced every STATIME

- PropertyName / VarName: Java variables and Java API settings, traced at the beginning of every file and at date changes.

Enhanced logging on MessageQueue protocol for throubleshooting

Enhanced logging on MessageQueue protocol for throubleshooting

At debug level 10 (supply details) the API traces additional information about each Message Queue supply, on Publisher and Subscriber side. This information can be used for debugging and throubleshooting.

The log traces include the MQ Status and the internal Action and Ack flags required by the Message Queue protocol.

Fixes

Fix for Message Queue race condition

A race condition could happen when creating a new Message Queue, when the application created and destroys Message Queues from different threads.

The race condition involves the un-publication of the Message Queue type, causing a new Message Queue with same type to be created as invalid.

Fixed.

Version 150: 02 December 2016

Summary

Recordset support, log timestamps with microsecond precision, memory usage stats after major GC event, ultra-low latency message passing for same-host connections

Enhancements

Memory Usage statistics after JVM Major GC event

The Java API now produces additional memory usage statistics after any Java Virtual Machine (JVM) Major Garbage Collection (GC) event.

Whenever a Major GC is completed by the JVM, the Java API reacts to this event and computes the instant memory usage referred to the Old Generation heap space.

This information is immediately reported to the ION Platform, stored and persisted into Daemon MONITOR logs. It can be leveraged by the ION Performance Meter tool to perform real-time alerting in case of critical memory shortage for Java components.

Log timestamps with microsecond precision (MiFID II regulation compliance)

The API is now able to generate accurate timestamps with microsecond precision on all the supported Operating Systems. Log timestamps with microsecond precision is also required to fulfill the requirements of the Markets in Financial Instruments Directive II (MiFID II) regulation for what concerns the ability to reconstruct all events related to an order throughout the lifetime of each order in an accurate time sequence.

To generate timestamps with microsecond precision, the API must make use of external libraries per the host Operating System.

For applications running on Linux/Solaris box the API must use binary .so libraries (already available since version 136).

For applications running on Windows box, the API must use a binary DLL library providing access to the OS time-stamp function available since Windows Server 2012. If running on older version of the Windows OS, the MiFID II compliance cannot be guaranteed.

All the above libraries are distributed as attachment to the Daemon product on the ION Tracker Website.

Refer to the document "Platform APIs - Programming Guide", chapter "ION APIs and MiFID II Regulation" for further details and to the document "Platform Tools: Installation and Migration Guide", chapter "End-To-End delays microseconds support - Linux and Solaris libraries for supported Java components" about how to install the required libraries.

New features and changes

Support for recordset

The recordset feature allows to make available on the ION Bus a single object (i.e a recordset) standing for multiple record objects, without the need to publish all records. This allows to reduce the memory footprint of the components (publication dictionary) and the network load.

The new `MkvRecordset` object has been introduced to support the recordset.

To benefit from the advantages of the recordset, both publisher and subscriber components need to use the new API.

Currently, the support for recordset is disabled by default. To properly configure the feature, the new settings "`mkv.publishonlyrecordset`" (publishing side) and "`mkv.userecordset`" (subscribing side) have been introduced.

For further information refer to the "Programming Guide".

Support for persistent subscription

The persistent subscription feature allows to subscribe records without worrying about the availability of the objects on the ION Bus at subscription time. The subscribed data will be provided when the related objects will actually be available on the ION BUS.

The API implements transparent resumption of the data flow after temporary unavailability of the data.

For further information refer to the "ION Programming Guide".

Ultra-low latency message passing for same-host connections

Starting from version 150, the ION Java API supports ultra-low latency data transfer among components running on the same host.

Using the following setting on both Publisher and Subscriber components:

- mkv.ultra_low_latency = 1 the connection among the two components running on

the same host will automatically switch from standard TCP/IP sockets to a proprietary message passing interface based on shared memory communication.

The ultra-low latency feature requires also the data producer and consumer being connected directly (i.e. data must not flow through the routing infrastructure).

Shared memory message passing can dramatically improve the performance of E2E latency between Publishers and Subscribers, breaching the microsecond for E2E latency.
For further information, please refer to the document "Platform APIS - Programming Guide" at "Chapter 5. Ultra-low latency with ION Java API".

Suppressing the SysMessage for low memory warning and error notifications

The Java API does not provide any more the warning and error messages related to memory running low.

Memory monitoring for Java application is demanded to the ION Performance Meter tool, using the Garbage Collection statistics (GC Stats and Major GC Stats) available through SysAdmin PerfMeter AddOn.

Fixes

NaN field always detected as changed

When a Publisher supplies a field with a NaN value, the supply goes out to all downstream components even though the value was already NaN and has not changed.

Now, the API fully supports NaN values and checks the internal representation to determine whether the value has changed.

Fixed.

Fix for ES Library local permission cache invalidation

A race condition on the ES Library could cause the local permission cache not to be invalidation in case of updates of Entitlement Server configuration.

This could result in query returning wrong permissions in case of concurrent configuration changes on the Entitlement Server.

Fixed.

Logtrace timestamp always in microseconds

Timestamps included in log traces are now always reported with microsecond precision, independently of USEMSECS configuration.

Version 138p3: 23 August 2016

Summary

Support for NaN supplies

Fixes

NaN field always detected as changed

When a Publisher supplies a field with a NaN value, the supply goes out to all downstream components even though the value was already NaN and has not changed.

Now, the API fully supports NaN values and checks the internal representation to determine whether the value has changed.

Fixed.

Version 138p2: 27 June 2016

Summary

Backward compatibility for date and time parameters in function call handlers.

New features and changes

MkvQoS supports the option to use date and time parameters in function call handlers.

Use the `MkvQoS.setPublisherUsesDateTimeObjects()` method if:

- The component is publishing an ION Bus Function with Date or Time parameters, and
- You want to receive an MkvValueDate or MkvValueTime object on the onCall() listener.

By default, the value of those parameters is received as plain integer objects.

Version 138p1: 16 December 2015

Summary

Fix for calling ION Bus functions with the required `DATE` and `TIME` parameters.

Fixes

Unable to call the Bloomberg BTS function BBG_UST_BloomSendHistoricalPrice(InstrumentID, Date,)

Java API components could not invoke ION Bus functions with the `DATE` and `TIME` parameters.

Version 138: 09 December 2015

Summary

Asynchronous supply notifications and a fix for calling ION Bus functions with DATE / TIME mandatory parameters

Enhancements

Asynchronous supply notifications

Java API version 137 and earlier can have synchronous, in-stack notifications for supply objects.

In-stack notifications for supply objects could happen in the following scenarios:

- On an extension of an existing subscription, that is, adding new fields. If the data dictionary contains field values that matched the new subscription field list, either partially or completely, the interesting data is notified synchronous, in-stack with the subscribe caller stack
- On a new subscription for an already subscribed record, within the same application, that is, using a different handler or different field list. If the data dictionary contains field values that matched the new subscription field list, either partially or completely, the interesting data is notified synchronous, instack with the subscribe caller stack
- On any subscription for a record that is published locally, within the same application. In these cases, a supply object is notified to all active local subscriptions, synchronous, in-stack with the supply caller stack.

Java API versions 138 and later suppress synchronous, in-stack notifications for supply objects.

All notifications for supply objects are performed asynchronously using a dedicated thread.

New features and changes

Fast failing for socket read errors

When an application connection is broken, the Java API now fails faster, detecting the read error from the socket as early as possible and contextually managing the associated socket closure and cleanup.

System message when connection is refused because of an Already Logged condition

The Java API issues a new system message when a connection is refused because of an Already Logged condition.

You can view and monitor this message using:

- The System Administrator Tool (SysAdmin) Event Viewer.
- The Performance Meter Tool (PerfMeter) and alert triggers. Here are examples

of the new message:

- `Login res refused 'comp1' -> 'comp2' ALREADY LOGGED.`

Or:

- `Login req refused 'comp1' -> 'comp2' ALREADY LOGGED.`

For more information about the ION monitoring tools, refer to the *Performance Meter: User Guide and Reference*.

System message because of consistent, repetitive connection failures

The Java API issues a new system message if there are consistent, repetitive failures when trying to establish a connection with a remote peer component.

If the connection failures happen due to a timeout, causing the connection to consistently remain in TRY status, the Java API uses the system message to report the condition to the ION Platform.

You can view and monitor this message using:

- The SysAdmin Event Viewer.
- PerfMeter and alert triggers. Here is an example of the new message:
- `Maximum number of connection failures detected: 'x' >= 'threshold'.`

For more information about the ION monitoring tools, refer to the *Performance Meter: User Guide and Reference*.

Fixes

Unable to call the Bloomberg BTS function BBG_UST_BloomSendHistoricalPrice(InstrumentID, Date,)

Java API components could not call ION Bus functions with the DATE and TIME parameters.

Applications can hang on shutdown

Applications using Java API version 137 can hang on shutdown, due to a server thread not properly stopped in time.

This has been corrected.

End-to-end delays and TickData distribution

The subscriber reports absolute end-to-end (E2E) delays that are not useful under the following conditions:

- When subscribing the data of a publisher linked with a C-API earlier than version 143 or a Java API earlier than version 131.
- When the USEMSECS configuration is different for the publisher and subscriber.

In these scenarios, the TickData distribution for absolute E2E latency could grow incrementally, causing a CPU disruption of the subscriber component.

This has been corrected.

Race condition on connection cleanup

A race condition could happen when cleaning up the connections towards a remote component.

The race condition triggered a NullPointerException during the connection cleanup, causing the connection to remain dangling.

When a subsequent connection attempt was made from the same remote component after its bounce, the Java API refused the connection with an Already Logged error.

This has been corrected.

Version 137p3: 01 December 2015

Summary

ES Library fixes:

- Profile rule conflicts resolved with `DENY` permission.
- Fix for potential exception during configuration download.

New features and changes

Conflicts on profile rule configurations at query time are solved with a DENY permission

For Generic by-Feature queries on profile-aware namespaces, all conflicts found in profile rule configurations at query time result in a `DENY` permission.

Conflicts are determined based on the parameters of the specific query.

Example 1 (conflict)

- User `mkv` belong to profiles `Trader` and `Viewer`.
- The following rules exist:
  - Rule_1:
  - Profile: `'Trader'` – Feature: `'Trade', 'ALLOW'` – Rule_2:
  - Profile: `'View'`
  - Feature: `'Trade/Edit', 'DENY'`
- The query for:
  - User: `'mkv'`
  - Resource: `'ANY'`
  - Feature: `'Trade/Edit'`

This configuration results in a `DENY` permission due to the conflict of Rule_1 and Rule_2.

Example 2 (no conflict)

- User `mkv` belongs to profiles `Trader` and `Viewer`.
- The following rules exist:
  - Rule_1:

– Profile: `'Trader'`

– Feature: `'Trade', 'ALLOW'` – Rule_2:

– Profile: `'View'`

– Feature: `'Trade/Edit', 'DENY'`

- The query for:
  - User: `'mkv'`
  - Resource: `'ANY'`
  - Feature: `'Trade'`

Results in an `ALLOW` permission because no conflict is detected for this query.

Fixes

Potential exception raised during Entitlements Server configuration download:

The Java API sometimes threw an exception c while downloading the Entitlements Server configuration data required by the ES library layer.

Version 137p2: 23 October 2015

Summary

- Porting of Java API 137 fixes.
- Fix for non-Daemon threads that were not stopped at shutdown.

Enhancements

Improved Java API performance when a custom thread changes system properties during the startup phase

According to the Java API startup model, customer threads should not change system properties until the startup phase is complete.

The Java API now accesses system properties in a more efficient way to manage cases in which customer applications do not respect a contract.

Fixes

Porting of Java API 137 fixes

Fixes present in Java API version 137 were not included in Java API 137p1, resulting in an inconsistency between the versions.

Now, all the fixes in Java API 137 have been ported correctly. This was a regression introduced in Java API 137p1.

Non-Daemon thread prevented the API from shutting down

All threads of the Java API component (`api_java`) are now correctly stopped when the API is shut down.

Version 137p1: 28 September 2015

Summary

- Improved Java API performance when a custom thread modifies system properties during the startup phase.
- Fix on stop.

Enhancements

Improved Java API performance when a custom thread modifies system properties during the startup phase

According to the Java API startup model, customer threads should not modify the system properties until the startup phase is complete.

The Java API now accesses system properties in a more efficient way to deal with customer applications not respecting this contract.

Fixes

Stop/Destroy methods did not stop the Java API

During the shutdown procedure, an internal thread was left running. This prevented the Java API from closing correctly.

For this reason, the process had to be forcibly stopped by an external signal to complete the close procedure.

This was a regression introduced in Java API 137.

Version 137: 29 April 2015

Summary

Entitlements Server library for local in-memory entitlement rule evaluations.

Enhancements

Integration of Entitlement Cache invalidation with External Entitlement Services

The Java API can now react to notifications of configuration changes sent by an External Entitlement Services.

On any notification, the Java API invalidates the local entitlement permission cache.

After a cache invalidation event, every new Entitlement check is evaluated remotely again using function calls to the relevant External Entitlement Service.

This mechanism requires the External Entitlement Service to have a status record published on the ION Platform, similar to the status record of the ION Entitlements Server. That is:

- The record name must use the following format:

  `EUR.ES_R_ENT_STATUS.<source>.<source>`
- The record should include at least the following fields:

– `Timestamp`
An integer field that reports the timestamp of the latest configuration change.

– `SeqId`
An integer field that reports the unique identifier of the latest configuration change.

– `DbName`
a string field that uniquely identifies the External Entitlement Service (it could be a symbolic string).

Improved documentation for MkvRecord class

Improved documentation for the following methods:

- `getLastTStamp()`
- `getLastMUTStamp()`
- `getLastMUTStampExt()`

New features and changes

Entitlements Server library for local in-memory entitlement rule evaluations

The Java API now includes the client-side library for Entitlements Server rule evaluations.

The Entitlements Server library automatically downloads the required configuration from the Entitlements Server. This configuration includes rules and namespace definitions for By- Feature Entitlements.

Based on the Entitlement configuration already downloaded, the Java API can execute local in-memory evaluations related to the following entitlement requests:

- Single entitlement request, using the following method:

```
MkvEntitlementManager::eval(
           MkvEntitlementRequest                    req,
           MkvEntitlementRequestEvalListener lner)
```

- Bulk queries for multiple resources, using the following method:

```
MkvEntitlementManager::bulkQuery( IMkvBulkEntitlementRequest
 req uest, IMkvBulkEntitlementResourceStreamFactory
                                     fac
tory,
String                                resourceFilter,
IMkvBulkEntitlementRequestListener    listener)
```

The bulk queries on multiple resources can be also performed initializing a Bulk Query Service that autonomously takes care of:

- Incoming function calls from remote clients
- Entitlement results delivery via Message Queue protocol. A new Bulk Query

Service can be created via the method:

```
MkvEntitlementManager::startBulkQueryService

(   String      name space,

IMkvBulkEntitlementResourceStreamFactory factory)
```

Delegation to an External Entitlement Service is also supported. This feature is transparent to the application.

For a detailed overview of the Entitlements Server library behavior, refer to the Java API

JavaDoc for the interface named `IMkvBulkEntitlementResourceStreamFactory`.

For an example of usage of the new interfaces of the Entitlements Server library, refer to the Java API Sample called `BulkEntitlements`.

Custom Stats for Entitlements Server library

The Entitlements Server library defines and produces Custom Statistics related to the Bulk Queries, including performance indicators and benchmarks.

The Custom Statistics can then be browsed using the ION Performance Meter (PerfMeter) user interface of the System Administrator Tool (SysAdmin).

Report for Generic ByFeature Entitlement Rules

The Entitlements Server library provides support for client-side creation of Entitlement Reports about configured ByFeature Entitlement rules.

The library supports in-memory reports for:

- A single namespace, including all the users on the ION Platform, using the following method:

```
MkvEntitlementManager::generateByFeatureRe
port(   String
                           entitlementName
space,
IMkvByFeatureEntitlementReportListener listener)
```

- A single namespace and a user, to get the report for the specific user, using the following method:

```
MkvEntitlementManager::generateByFeatureRepor
   t(   String
                              entitlementName
   space,
   String                                 userName,
   IMkvByFeatureEntitlementReportListener listener)
```

- A single namespace, a user, and a feature, to get the report on the specific feature for the specific user, using the following method:

```
MkvEntitlementManager::generateByFeatureRepor
   t(   String
        entitlementName space,
   String     userName, String    feature,
   IMkvByFeatureEntitlementReportListener listener)
```

For a detailed overview of the Entitlements Server library behavior, refer to the Java API *JavaDoc* for the interface called `IMkvByFeatureEntitlementReportListener`.

For an example of usage of the new interfaces of the Entitlements Server library, refer to the Java API Sample called `ByFeatureReport`.

New MkvUnicodeBuffer utility class

The utility class `MkvUnicodeBuffer` has been added. This class implements a mutable char sequence for fast string management.

For more information, refer to the *JavaDoc* of the Java API.

Fixes

Fix for MkvSupplyTimestamp class

When the `USEMSECS` setting was set to `0`, the Millisecond Unit Timestamp value was not computed correctly.

This has been corrected.

Fix for NullPointerException in MkvSub Persistent Subscription of Chain

When a record was deleted from a chain, there could be a NullPointerException in the MkvSub persistent subscription of the chain.

This has been corrected.

Version 136p2: 24 February 2015

Summary

- Change in types handling with duplicate field names.
- Fix for potential crash at start up.

Enhancements

Prevent creation of MkvType with duplicated fields

An `MkvType` object is an ordered list of `{name, type}` pairs shared between records referring to the same business objects.

Individual record fields of `MkvType` objects can be accessed by numerical index or field name.

A publisher declares the ordered list of `{name, type}` pairs as an argument for the constructor of an instance of the `MkvType` class.

Prior to version 136p2, the `MkvType` class constructor accepted list of fields with duplicate names.

When there are duplicate names, only one of the same-named fields is accessible.

This can lead to application errors. For example, subscriber applications may stall waiting for delivery of field values that the publisher does not provide.

The `MkvType` constructor now fails throwing an exception when provided a list of fields with duplicate names, enabling the implementation of fail-fast behavior on the publisher side.

The new behavior will break publishers that relied on the behavior implemented in older versions.

Publisher applications linking against the new Java API version must be reviewed to ensure that no duplicates field names list is provided as arguments to the `MkvType` constructor.

The behavior of the subscriber-side of previous Java API versions has been retained. The Java API still accepts types with duplicate field names in case these have been originated from a remote component.

Types with duplicate fields will be reported in the PSH logs file at any debug level.

Fixes

Fix for potential crash at startup

Starting with version 136, Linux and Solaris builds of the Java API implement high resolution timestamps in native code for improved accuracy.

The original implementation of the mechanism exposed a Java Development Kit (JDK) bug that resulted in the Solaris runtime loading the Linux shared object. This caused the Java Virtual Machine (JVM) to crash at startup.

Fix for chain idle on timeout

In case of slow connectivity, a spurious chain idle event sometimes fired during the initial chain download.

Moreover, no notification was issued on the download end.

Version 136p1: 2015-01-28

Summary

- Using Daemon Master time for custom stats
- New method to report custom end-to-end (E2E) delays based on supply

Enhancements

New method to report custom E2E delays based on supply

All components reporting custom E2E delays based on `MkvRecord` supplies should now rely on the new `MkvE2EDelaySampler` interface:

```
public void sampleSupplyTimestamps(long timestamp, double timestampExt);
```

The standard usage of the function shown above is by passing the supply timestamps to the custom E2E delay. For example:

```
 MkvE2EDelaySampler  sampler  =  mkv.getPlatform().getE2EDelaySampler(peer);
 if(sampler == null) sampler = mkv.getPlatform().createE2EDelaySampler(peer);
sampler.sampleSupplyTimestamps(record.getLastTStamp(),
record.getLastMUTStamp());
```

Version 136: 19 December 2014

Summary

- Custom statistics for E2E delays.
- MkvTickData API.
- Improved memory performance of logging.

New features and changes

Custom Stats for End-To-End Delays

All components using Java API version 136 report custom statistics for E2E and application round-trip delays.

Such delays are reported in milliseconds, with microsecond granularity.

The custom statistics can be consumed by the ION Performance Meter Tool (PerfMeter) and monitored using the PerfMeter user interface or the ION System Administrator Tool (SysAdmin).

MkvTickData API

Applications using Java API version 136 can capture, collect, and report latencies or other performance measure in a tick-by-tick way using the new `MkvTickData` API.

The `MkvTickData` API serializes the collected statistics on a per-second basis into the `STATISTICS` log file of the component. The `MkvTickData` API can also calculate frequency and percentile on the capture measures.

The chosen measures should be registered to the Java API, either at startup or dynamically, during component execution. For example:

```
MkvE2ETickData customDelayTicks = new MkvE2ETickData("Market Delay
Absolute", "MDATA", "BBG_MDATA");

Mkv.getInstance().getTickDataCollector().register(customDelayTicks);

MkvTickData cpuTicks = new MkvTickData("CpuAvg", "BBG_MDATA");

Mkv.getInstance().getTickDataCollector().register(cpuTicks);
```

New measurement ticks can be captured by the application using the `add()` method on the pertinent `MkvTickData`:

```
customDelayTicks.add(currentDelay);
```

For more information, refer to the Java API *JavaDoc* or to the `MkvTickDataApi` application in the package of samples.

Enhancements

MkvE2EDelaysSampler

The Java API for reporting custom, application-level E2E delays has been enhanced.

ION recommends using `sampleTimestamp()` and `sampleDelay()` methods for reporting custom E2E and custom application round-trips, respectively.

The input parameter should be the value of the most precise available system timer, in milliseconds, computed as returned by the `Mkv.currentTimeMillis()` and `Mkv.milliTime()` functions for E2E and application round-trips, respectively.

Such delays are reported in milliseconds, with microsecond granularity.

Improved performance for logging

A new interface is available for Java API logging, to log a `CharSequence` and provide better performance with respect to the garbage collection (GC) and the young memory generation.

135: 10 September 2014

Support for generic auditing and fixes.

New features and changes

Support for generic audit logging

Version 135 introduces the Generic Auditing API to deliver audit messages to a centralized store. For more information, refer to the following:

- *JavaDoc* of `com.iontrading.mkv.MkvPlatform.logGenericAuditEvent`
- *Auditing Service: User Guide and Reference*

Resiliency log file for the Generic Auditing API

When a component uses the Generic Auditing API, all calls to the external Auditing Server are traced to the new log file named `AUDIT_JOURNAL`. The traces are dumped in a format that can be replayed by the ION Data Player component (DPlayer).

The file is generated by default. The following setting can be specified in the component `mkv.jinit` to force the Java API to disable the resiliency file:

```
mkv.audit_journal = 0
```

Configuration setting for Generic Audit Source

The following configuration setting is available to set the source of the external Generic Auditing Service:

```
mkv.audit_source
```

The default value is `AUDIT_SERVER`. This value is equal to the default source of the ION Auditing Server component.

Enhancements

DRecorder module enhancement

The DRecorder module embedded in the Java API library can now report queue events and subscriptions.

The enhancement allows the DRecorder of the Java API to provide the same functionality as the C-API.

Version

Summary

Notification of pending function calls failure on function unpublishing

Starting with version 135, when a function object is unpublished, all calls pending response for that function are issued with an error.

For more information on this enhancement, refer to the compatibility information section.

Fixes

Component Deadlock

A bug in the data entitlements subsystem sometimes caused a deadlock, preventing the component from accessing the ION Bus.

DRecorder traces fix

The DRecorder scripts did not report function object publish events.

Furthermore, there were some inaccuracies, such as switched PEDIT on/off traced into the DRecorder log and incorrect peers reported in publisher side subscription traces.

RunningValue of drecorderscripts variable shown as [not set]

SysAdmin displayed `RunningValue` of the `drecorderscript` as `[not set]` even if the `ConfiguredValue` was configured correctly. This has been fixed.

Missing chain data in MkvPersistentChain subscription

The bug was related to scenarios in which the set of records belonging to the chain after an edit sequence was not a superset of the original set (that is, some records were removed besides others being added).

The issue occurred only when subscriptions were performed using the `MkvPersistentChain` wrapper.

This has been fixed.

133p7: 18 June 2014

- `MkvDate` utilities.
- Fix to prevent registration on two different daemons simultaneously.

New features and changes

MkvDate utilities

A new static class providing timestamp utilities has been implemented.

Fixes

Idle is not sent when publishing has been stopped

A spurious idle message was sent out on logon even when publishing had been disabled by a previous invocation of the following:

```
MkvPublishManager.publishStop()
```

Prevent registration on two different daemons simultaneously

When a registration-success message from a previously non-responding Daemon was received after switching registration to another Daemon, the component would appear as if registered on two different Daemons. This caused the component to work incorrectly.

The Java API has been fixed to drop the stale registration message.

Version 134p2: 07 February 2014

Summary

- New accessor for the user type.
- New setting to disable logging.

New features and changes

New Mkvuser.getType() method

The Java API now provides an accessor for the following user types:

Version

Summary

- Admin
- Publisher
- Viewer

New mkv.mindiskfree setting

This new setting enables users to disable logging if the available disk space gets below a specific threshold.

This value is expressed in MB. The default value is `0`, which indicates no threshold is set.

The Java API reports log suspensions and resumptions events as platform errors and platform warnings, respectively.

Version 133p6: 20 September 2013

Summary

Fix to prevent exceptions in the MkvSub implementation.

Fixes

Issue ID IONCS-251043: IndexOutOfBoundsExceptions raised from Java API when using MkvSub

An `IndexOutOfBoundsException` was raised internally and caught by the Java API in a specific use case involving the chain being unsubscribed within the context of a listener registered within the MkvSub interface.

This was due to an error that could further result in spurious/missing notifications on that chain on following subscriptions.

Issue ID IONCS-251560: ClassCastExceptions raised from Java API when using MkvSub

A `ClassCastException` was raised internally and caught by the Java API when a chain that had been previously subscribed using `MkvSub.addChain` was explicitly unpublished by its owner.

No exception was caused by unpublish events due to disconnections. Other than being logged, the exception had no functional side effect.

134: 14 June 2013

- UTF-8 encoding support for field values.
- Chain indexing.
- Various fixes.

Enhancements

Optimized chain lookups

Chains implementation has been enhanced with an index. As a result, lookup time is now approximately constant rather than linear.

Fail-fast approach in handling incorrectly formatted component key

When a syntactically invalid key is provided, `Mkv.setComponentType` immediately throws an exception rather than fail silently without attempting the registration.

New features and changes

UTF-8 encoding support for field values

Strings containing characters outside of the ASCII range can now be provided as values for record fields and function parameters.

The feature is fully compatible with existing routing infrastructure. However, it requires that all Java API-based components involved in producing and consuming the data are built with a version of the Java API supporting it.

C components can interoperate with components implementing this feature if they provide and consume zero terminated UTF-8 encoded byte sequences for string values.

Report actual Garbage Collection duration in addition to other Garbage Collection statistics

Garbage Collection (GC) statistics have been enriched by providing cumulative CPU time as provided by `sun.gc.collector` counters.

Version

Summary

Report threads IDs in log files

Each individual line in the log files reports the identifier for the thread that invoked the logger.

The IDs can be disabled using the following configuration:

`mkv.applicationlogsthreadid = 0.`

Report timestamp in stderr.log

Each line in the `stderr.log` file now reports the trace generation time.

Improved tracing of object identifiers

Traces involving message exchanges over the ION Bus (for example, delivery of subscriptions/supplies) have been enriched with the numeric identifiers by which the relevant ION Bus objects are referenced.

Furthermore, new traces for binding of identifiers to objects have been introduced. These latter traces are enabled for `mkv.debug <= 30.`

New traces have been introduced to report failed object look-up (for example, when a supply is received for a record that has been unpublished).

Unique chains feature

User can now create instances of chain objects with a uniqueness attribute. Chains

created this way reject addition of duplicate elements.

Line length reported in log files

Log file line headings have been extended to report the actual number of bytes before the next line separator. This enables easy parsing of the logs when a single logical line is broken because of exception stack trace reports.

Fixes

Full snapshot was not provided after extending a subscription

The `MkvSupply` object provided as the value for `MkvRecordListener.onFullUpdate` always provides the full set of fields subscribed.

Previously, this was not the case for the snapshots received in response to an extension of an existing subscription.

Java API date type inconsistent using getValue("Date").getType() instead of getMkvType().getFieldType("Date")

`MkvValues` returned by the Java API now reflect the original field type used in the object type declaration.

For example, `getValue("Date").getType()` returns `MKV_DATE` instead of `MKV_INT` if field `Date` has been declared as a date type.

Missing notification of IDLE event for empty chain subscriptions

The bug occurred when an empty chain was subscribed after the timeout-based idle had been received for another subscription on the same chain.

Failure of DayEnd log rolling

A clock drift issue could result in the active log not being rotated at midnight. This could cause traces created at a later time to be delivered to the log file for the day before.

Fixed socket descriptor leak on secure connections

A single descriptor was leaked every time a secure connection was initiated. This could have an impact in case of multiple failures due to registration problems (for example, if an incorrect password was provided).

Null Pointer Exception being reported in logs on closing a message queue subscription

This error sometimes occurred when the message queue publisher concurrently unpublished the queue.

However, outside of the exception, there were no other side effects for the race condition.

Version 133p5: 11 June 2013

Summary:

Fix for exceptions raised when using the MkvSub interface.

Notes

The locking mechanisms have been revised to prevent a few data corruption issues that affected previous Java API versions.

In some specific cases, the current implementation involves notifying user-level listeners while holding internal locks. To prevent deadlocks, user code running in the context of a listener should release any locks before invoking methods from the Java API.

Fixes

ConcurrentModificationException when using MkvSub api (Issue id.: IONCS-248675)

Due to a missing synchronization construct, which is now implemented, a `ConcurrentModificationException` was thrown when the `MkvSub` API was used.

Missing notification of IDLE event for empty chain subscriptions

The bug occurred when an empty chain was subscribed after the timeout-based idle event was received for another subscription on the same chain.

Race condition in notification of local chain snapshot could result in missing notification of chain events

Synchronization has been added so no change can occur during the notification of the local chain snapshot.

Race condition in notification of local record snapshot could result in record being in an invalid state

Synchronization has been added to enforce ordering and data consistency during notification of local record snapshot.

Download of chains published by the Daemon component would take a significant amount of time

This issue only affected chains published by the Daemon component. It happened was due to the chain download not timing out as specified by the `mkv.cidletimeout`.

The implementation has been revised to complete the download at most within the specified limit.

Non-atomic update of filtermask related to subscribemask could result in missing field updates

This problem occurred when multiple threads changed a subscription on the same record.

MkvSub.addRecord, MkvSub.addPattern, MkvSub.addChain, MkvSub.addRegExp and MkvSub.addGlob had no effect when issued after MkvSub.start

The implementation has been fixed so that changes in the interest set are immediately picked up after MkvSub instance has been started.

Version 133p4: 15 May 2013

Summary:

Fix for deadlock when subscribing local MkvChain objects using the MkvSub interface.

Enhancements

Enforce 256 characters limit on ION Bus object name size on creation

This enhancement limits the creation of the object with a name longer than 256 characters. The behavior is now consistent with the one of the C-API.

New MkvMQ.putUserAction overload

A new method has been introduced that allows passing only one parameter when it is not required to provide supply data as well.

New features and changes

com.iontrading.mkv.MkvEntitlementManager.getServiceStatus() method is now public

The method `com.iontrading.mkv.MkvEntitlementManager.getServiceStatus()` has been changed to public from this version.

Fixes

Issue ID IONCS-245525: Deadlock when subscribing MkvChain objects using MkvSub interface

The deadlock dealt with a race condition involving concurrent read/write access to a chain. The probability of triggering the deadlock increases when the chain is a local object.

Missing validation of runtime settings

Configuration settings sent by a System Administrator were not validated by the API. Therefore, it was possible to inject out of range values for numeric properties.

Fix for race condition at download disabling subscriptions

A race condition during download could result in an exception tearing down the data entitlement event handling stack. Thus, the application could fail to subsequently carry on subscriptions.

Issue ID IONCS-229538: Fixed socket descriptor leak on secure connections

A single descriptor leaked every time a secure connection was initiated. This could have an impact in case of multiple failures due to registration problems (for example, if a wrong password was provided).

MkvSupplyBuilder setField stopped accepting nulls

When an object is null, `MkvSupplyBuilder.setField(int/string, Object)` now is the same as `MkvSupplyBuilder.unset(int/string)`.

This is a regression introduced in version `api_java` 130 that is now fixed.

Null pointer exception thrown when adding fields to a persistent subscription

A Null Pointer Exception was thrown on extending the field set for a `MkvPersistentRecordset` instance obtained using the `MkvSubscribeManager(MkvRecordListener)` method.

This bug is now fixed.

Version 133: 25 October 2012

Summary

Support for a negotiated shutdown protocol and various enhancements on message queues.

Enhancements

Users can now retry answering the functional call with a shorter reply, or with an error

The function call context is not destroyed when a *supply too large* exception is raised.

New features and changes

New Message Queue (MQ) custom statistics

The Java API now broadcasts the following running counters, which are specific to message queues on the MQ Stats custom table:

- `MQPubLocal`
- `MQSubLocal`
- `MQPut`
- `MQSup`

Message queue subscriber side application driven acknowledgement

A message queue subscriber can register a listener on acknowledgement requests from the queue publisher. After the request is received, delivery of messages will be suspended until explicitly resumed by invoking the `MkvMQSubscribe.replyToAckRequest()` method.

This way, a subscriber can postpone acknowledgement of events until it has actually consumed them.

Support for the in-order delivery of application level protocol messages over message queues

The new `PSH_MQACTION_USER` action has been added to support in-order delivery of application-level protocol messages over message queues.

For use case descriptions, refer to the *Programming Guide*.

Negotiated component shutdown logic

A component is notified of a `PSH_EVENT_SHUTDOWN_REQUEST` event immediately before receiving an external stop request.

By invoking the new `Mkv.shutDown (MkvShutDownMode` mode (string message) with the `MkvShutDownMode.ASYNC` mode, the component can now postpone shutdown to carry out a lengthy tear-down activity.

If this feature is supported, a controlling Daemon is notified of the deferral and act accordingly.

Message queue publisher interface to allow asynchronous notification of flush events completion

The `MkvMQ.flushEx` method provides a way for a message queue publisher to be notified after all previous queued items have been acknowledged by the message queue subscriber.

This method can be used with subscriber driven acknowledgments to implement reliable application level delivery of message queue items.

Obtain a list of all records available for a given type

The new method `MkvPublishManager.getMkvRecords` (string type) returns a list of all records available for the given type.

Fixes

Wrong notification of a full update event for records subscribed by pattern

In some cases, a bug resulted in missing fields in the `SupplyObject` associated with first `onFullUpdate` invocation for record updates received on behalf of a pattern subscription.

IONCS-224097: Fix for race condition resulting in missing type for message queue

This happened when a queue with a shared type was published at the same time the last queue for that type was being destroyed.

IONCS-223633: Race condition when un-publishing results in a queue type loss

Same as issue IONCS-224097.

Version 132: 10 July 2012

Summary

New data recording functionality.

Enhancements

CUSTOM.STATS log file is no longer created by default

Creation of the `CUSTOM.STATS` log file is now controlled by the `mkv.logcustomstats` configuration variable

The file is generated when `mkv.logcustomstats` is set to 1. The file is not generated when the variable is set to the default value `0`.

New features and changes

Data recording functionality

The Java API now supports the data recording functionality. This functionality enables the API to create recordings of data received using publish-subscribe protocol for later replay using the ION Data Player tool (`dplayer`).

This functionality is enabled by setting the `mkv.drecorder` configuration variable with the recoding file name.

The `mkv.drecorderextobj` and `mkvdrecorderownobj` configuration variables can be used to enable recording for external and local objects, respectively.

For further details, refer to the *Java API Reference Guide*.

Fixes

Java API was not able to subscribe to Daemon-published objects after unregistration

The issue had to do with publish objects not being removed from a Java component's cache when the component was disconnected from the Daemon.

On reconnection, the component was not be notified of the republished objects. Therefore, the component was unable to re-subscribe them. As a result, the component did not receive any further updates for these objects.

Creation of message queue instances leaked Class Object to Permanent Generation

The internal implementation used to dynamically create code for implementing internal proxy objects (which were subsequently cached) resulted in Class Objects getting stuck in the Permanent Generation until the queue filled up.

Version 131p2: 21 April 2012

Summary

Extended Garbage Collection (GC) stats, support for `user.init` credentials, and fixes.

Enhancements

Round trip times now expressed in usemsecs units

Round trip time reported in statistics are now expressed based on the `usemsecs` setting.

Extended Garbage Collection statistics

In addition to the usual aggregated memory statistics, all Java components now publish the following additional statistics related to GC activity.

Table 2: Statistics related to GC activity

Statistic	Description
`YoungGenUse`	Heap usage in KB of the young generation.
`YoungGenMax`	Maximum heap usage in KB of the young generation since startup.
`OldGenUse`	Heap usage in KB of the old generation.

Statistic	Description
`OldGenMax`	Maximum heap usage in KB of the old generation since startup.
`PermGenUse`	Heap usage in KB of the permanent generation.
`PermGenMax`	Maximum heap usage in KB of the permanent generation since startup.
`MinCollCnt`	Number of minor collections in the statistics time slot.
`MinCollDur`	Cumulative duration in milliseconds of minor collections in the statistics time slot.
`MajCollCnt`	Number of major collections in the statistics time slot.
`MajCollDur`	Cumulative duration in milliseconds of major collections in the statistics time slot.

New features and changes

Support the ION user_dmp tool decryption algorithm

The new `MkvGatewayHelper` class enables the validation of credentials encrypted in the format supported by the ION `user_dmp` tool.

Method for retrieving MQ internal cache size

The `MkvMQ.getSize` method returns the site of unacknowledged queue updates. This can be used to drive control flow at the application level.

Fixes

IONCS-191362: Incorrect RTD statistics on Microsoft Windows platform

On Microsoft Windows platforms, the coarse timer granularity can result in Round Trip Delay (RTD) values lower than 15 milliseconds being accounted for as 0 milliseconds.

Version 131p1: 30 January 2012

Summary

Extended and revised Custom Stats examples.

Enhancements

New code samples for Common Market statistics and Virtual Hosts

Examples show how to use the Custom Stats interface to report Common Market statistics. The examples and supporting scripts are bundled with the Java API package.

Version 131: 30 September 2011

Summary

Support for secure registration connection.

Requirements

To support the secure registration channel and the Custom Stats feature, Daemon Version 106p4 or later is required.

Enhancements

Example of External Authentication component

The examples package now includes sample code illustrating how to perform External Authentication.

New features and changes

Secure registration channel

Authentication on the ION Platform and general communication with the Daemon is now secured using a registration channel based on the TLS protocol.

CustomStats interface for declaring custom statistics tables

A new mechanism for reporting custom statistics to the ION Platform has been added. For more information, refer to the documentation and examples package available in the Java API bundle.

Version 129p2: 13 September 2011

Summary

Various fixes, including a fix for missing queue updates.

New features and changes

Users can now get the name of the latest Daemon to which the component successfully registered

The new method `Mkv.getLastRegistrationDaemon` returns the name of the last Daemon to which the component successfully registered.

Fixes

Record image was no longer available when persistent chain listener was notified of the removal of a record

This issue occurred because persistent chain subscription first removed the record subscription, then notified listener.

Therefore, at the time when the listener was notified that the record had been removed, the record image was no longer available.

Platform user attributes were not correctly updated when credentials were changed at runtime

In certain cases, changing a user's password resulted in a component no longer being able to accept inbound connection from other components registered in with that user.

Platform user could not be retrieved from component object

When registered to a Daemon Slave, a component would not pick up new users created after the component had been started.

Consequently the component was unable to retrieve the Platform user from components registered with the newly created user.

Missing queue updates

If a publisher component implemented on the C-API added elements to a queue in short batches, the subscriber could get stuck and receive no further updates.

This bug only affected subscribers where the publisher components were implemented using the C-API.

Fix for Exception in user-defined listener (Issue ID.: 00169327)

The exception was raised because of a missing synchronization construct in an internal subsystem.

Version 130: 26 May 2011

Summary

Support for Generic By-Feature Entitlements and record image persistence.

New features and changes

Generic By-Feature Entitlements

This version supports the Generic By-Feature Entitlements mechanism available in the Entitlements Server Version 105 onwards.

Generic By-Feature Entitlements allow users to overcome the limitations of the standard By- Feature solution and provides greater flexibility in configuring the system. Entitlement rules can be configured for a generic hierarchy of resources, rather than bound to the usual fixed four-levels of ION Platform identifiers (Type, Container, Source, and Object).

The new API calls allow:

- Checking the entitlements using the Generic By-Feature mechanism.
- Downloading the complete set of entitlement rules configured in the Entitlements Server, with specified filters.

For more information, refer to the *Entitlements Server: User Guide*.

Record image persistence

A new subscription interface has been added that supports the concept of configurable subscription objects. Such subscriptions can be configured as *cached*, that is, to persist an image of records that are no longer available.

In addition to supply data events, notifications of life-cycle events are provided for objects subscribed using the new interface.

The new functionality is provided by interfaces in the `com.iontrading.mkv.mkvsub package.`

The entry point for the service is the `MkvSubManager` class, which can be obtained through `Mkv.getInstance().getMkvSubManager()`

Fixes

MkvRecordListenerExt visibility fixed

The `MkvRecordListenerExt` interface now has public visibility.

Version 129p1: 03 March 2011

Summary

Fix for missing queue updates and other fixes.

Fixes

Fix for missing queue updates (Issue ID: 00160648)

The fix solves a regression with respect to Version 127 that affected subscribers of publisher components implemented on C-API.

If such a publisher puts elements in a queue in short batches, the subscriber can get frozen and receive no further updates.

Fix for missing chain idle

If:

- An empty chain was subscribed, and
- A record was appended to the chain before the CIDLETIMEOUT expiration.

Then:

- No chain idle was notified.

This affected client applications relying on receiving at least one *chain idle* event after subscribing a chain.

Fix for missing process identifier in statistics info (Bug ID: 77322)

The process identifier was not:

- Returned correctly by the Java API
- Displayed in the PID field of the Platform Components view of the System Administrator tool

Furthermore, the PID information was not traced in the PInfo statistics.

Null pointer exception raised when the license function from Daemon / CS was not available:

A null pointer exception was raised if the license function from the Daemon / CS was not available.

Now, an explicit and meaningful exception is thrown when this happens.

Spurious chain set event

In some cases, a chain idle event could trigger a spurious chain set event.

Version 128p5: 25 February 2011

Summary

Missing queue updates and other fixes

Fixes

Missing queue updates (Issue ID: 00160648)

The fix solves a regression with respect to Version 127 that affected subscribers of publisher components implemented on C-API.

If such a publisher puts elements in a queue in short batches, the subscriber can get frozen and receive no further updates.

Fix for missing chain idle

If:

- An empty chain was subscribed, and
- A record was appended to the chain before the CIDLETIMEOUT expiration.

Then:

- No chain idle was notified.

This affected client applications relying on receiving at least one *chain idle* event after subscribing a chain.

Null pointer exception raised when the license function from Daemon / CS was not available

A null pointer exception was raised when the license function from the Daemon / CS was not available.

Now, an explicit and meaningful exception is thrown in that case.

Spurious chain set event

In some cases, a chain idle event could trigger a spurious chain set event.

Version 129: 12 October 2010

Summary

- Enhancements to Message Queues
- Support to Entitlements Server 103

- Other improvements and fixes

Enhancements Message Queues

Notify queue on destroy

The application can now register a listener that is notified when the queue is closed. When this event occurs, an iterator is provided to access the queue cache.

Disable Message Queue timeout on unsubscribe

A message queue can be set up with an associated timeout. When the timeout expires, the message queue is closed.

Users can now prevent the message queue from being closed when the timeout expires. To do this, set the `DisableTimeoutonUnsubscribe` flag.

No need to provide supply object if only action field is to be provided on message queue update

The Message Queue now supports the following new method:

- `MkvmessageQueue.put`

This method provides a dummy supply. With this method, users are not required to provide a supply object when only the action field is to be provided.

Improved recovery of Message Queue subscription after disconnection

Message Queue updates are now immediately delivered after client re-connection.

Message Queues performance improvement

Message Queues are implemented using publish / subscribe protocol. To improve performance, the standard control flow mechanism is disabled during the Message Queue implementation.

No Coalescing of Message Queues

Publishers no longer need to flush a message queue periodically to ensure prompt delivery of data.

Others

MkvFieldNotFoundException class extended

The `MkvFieldNotFoundException` class supports a new method:

- `getField`

This method is used to retrieve the name for the missing field.

The action code can be now sent without providing a dummy supply object.

Support for low latency messages

The new method `MkvPlatform.endFlushBlock` may prompt the Java API to immediately send a message to the socket when possible, short circuiting the default message sending mechanism. This can delegate the sending of messages to a different thread.

This is an advanced feature specifically addressing programs with low latency requirements. A new configuration setting is available:

- `mkv.dtsyncsend`

New features and changes

On behalf of registration and authenticated license management:

A component can now:

- Register *virtual components* on its behalf, and
- Acquire and release licenses accordingly. For details, refer to the Java API documentation.

Support of complex entitlements queries:

The feature allows users to query the Entitlements Server with complex criteria for immediately downloading entire lists of entitlements.

For details, refer to the Java API documentation.

New methods have been added to extend the field set for persistent record and chain subscriptions:

The following methods allow users to extend the field set for persistent subscriptions for both records and chains:

- **For records:**
  MkvPersistentRecordSet.subscribe
- **For chains:**
  MkvPersistentChain.subscribe

Expose ThreadFactory delegators in MkvQoS

A new feature has been added to allow a client application to register its own ThreadFactory in the API.

This is an advanced feature. For details, refer to the Java API documentation.

Enhanced Support to Entitlements Server 103

The Java API now supports the Entitlements Features introduced in Version 103.

For details, refer to the Entitlements Server 103 release notes.

An extended MkvRecordListenerEx interface has been added

The unpublish event of a record can now be managed by a listener specific to the unpublished record, rather than by the global listener.

Notifications of unpublish events for persistent subscription

The application is notified about the unpublishing of records that have been subscribed using the persistent subscription mechanism if the listener implements the `MkvRecordListenerEx` interface.

New methods for calling transactions / functions getting the identifier at the same time

Two new methods have been added:

- `MkvFunction.callWithId`
- `MkvRecord.transactionWithId` They have the same behavior as:
- `MkvFunction.call`
- `MkvRecord.transaction`

However, they also return to the caller the identifier of the function call / transaction request.

Fixes

Fix for race condition when record is updated at the same time congestion condition ends

Records updates were sometimes delivered in the wrong order because of a race condition. This happened when a record was updated at the same time a congestion condition ends.

MessageQueue type was not unpublished when MessageQueue is closed / destroyed

Now when the message queue is unpublished, its type is also unpublished if no other queue exists for the same type.

Version 128p4: 11 October 2010

Summary

- Persistent chain subscription improvements
- Enhanced tracing and statistics
- Deadlock detection and reporting

Enhancements

Detailed trace of Garbage Collection activities

Users can now configure the tracing activity of the Java Garbage Collector

(GC). To do this, use the `mkv.gctrace` variable, setting the variable to:

- All = (Default) The GC traces any events
- None = GC tracing is disabled
- Old = GC traces only activity affecting the old / tenured heaps

Deadlock detection and reporting

When exiting, the Java API checks whether a deadlock has happened. If it has, it dumps a stack trace onto the PSH logs.

Configurable internal chain idle time out

A new `CIDLETIMEOUT` configuration parameter allows users to set a timeout for delivering an idle event in the following cases:

- After a chain subscription to legacy publishers, and / or
- After a subscription to a chain with no records.

Timestamp information added to stderr.log file

Startup timestamp and version number are now included at beginning of the `stderr.log` files.

Fixes

Fix for a potential deadlock

Two possible causes of deadlock have been detected and fixed. The

deadlock could occur in the following scenarios:

- Unsubscription of a chain as result of a disconnection and concurrent subscription of the same chain from another connection
- Unsubscription of a pattern and concurrent re-publish of a type containing pending records (that is, records that need to be re-published together with the type)

Deadlocked custom component (Issue ID: 00135214)

In very rare circumstances, a deadlock could happen during the login phase, making the component unable to manage further connection events.

Problem with editing of CS-published types

Editing a type published by the CS without changing it prevented any further editing of that type.

Fix for wrong SubRecOut statistics when publisher disconnected

The SubRecOut statistics sometimes provided an incorrect value when the original publisher was disconnected from the routing path.

Fix for missing type at download (Issue ID: 00133446)

Due to a race condition, types were sometimes lost when Router_M was:

- Downloading publication types from a newly connected component, and
- Delivering publication types to newly connected components.

Persistent Chain Subscription Performance Improvement (Issue ID: 00141302)

When performing persistent subscriptions to large chains (around 200K records), the download can hang. This can result in a timeout, which will cause the download to fail.

The download implementation has been optimized to prevent the download from hanging and therefore causing the timeout.

Version 128p3: 14 May 2010

Summary

- Fix for objects not published between a start and stop event.
- Other minor fixes and changes.
- Performance improvements.

Enhancements

New methods to convert timestamps

The following methods have been added:

- `MkvSupplyUtils.getTimestamp`
- `MkvSupplyUtils.setTimestamp`

These methods convert between ION integer timestamp representation and Java Calendar and Date objects representation.

Additional information in the log traces

The following information has been added in the `Pinfo` trace:

- API version
- Application version
- Java Virtual Machine (JVM) version

Relaxed MkvSupplyFactory.create requirements

`MkvSupplyFactory.create(Object[])` can now pass a vector of any supported type (Integer, String, Double, MkvValue).

Previously, only a vector of the type Object could be passed.

Performance improvements in MkvStringBuffer

Performance improvements allow faster string copy and `MkvPublish` cache look-up.

New features and changes

The Mkv.destroy method is now available

The `Mkv.destroy` method can be called on the `Mkv` class instance to shut down the Engine. This can be useful when the Engine is running within an application server.

Low-level traces about network activity can be now enabled

The `TRACE_DT` variable can be set to enable low-level traces regarding network activity. Values:

- `0` =Tracing disabled (default).
- `1` =Tracing enabled for all connections.
- `-1` =Tracing enabled only for the connection with the CS / Daemon.

Propagate errors through message queue

It is now possible to propagate out-of-band errors by:

- Calling the `MkvMQ.setError` method on the publisher side.
- Intercepting the `MkvMQSubscribeAction.ERROR` event on the subscriber side.

Fixes

Objects published between a publish stop and start were not actually sent over the network (Issue ID.: 00114540)

If a publisher had published objects between the invocation of the `MkvPublishManager.publishStop` and `MkvPublishManager.publishStart` methods when no component was connected to it, the published objects were not sent over the network when remote components were connected at a later stage.

Fix for minor memory leaks

Minor memory leaks have been detected and fixed in:

- The statistics generation procedure.
- The implementation of the `MkvPersistentChain` and `MkvPersistentRecordSet` classes.

Fix for connection statistics information

The `SubRecIn` value of the `CInfo` statistics trace was not decremented as expected when records were unsubscribed.

Back-compatibility fix for component implementing data entitlements with old Daemons / CS

Entitlements support required the API to wait for the `IDLE` event from the CS / Daemon before allowing any subscription.

CS and Daemon versions prior to 132p3 and 102p1 respectively do not send the `IDLE` event and the API does not ultimately allow subscriptions to be performed. This causes the component to become blocked.

API Version 128p3 works around these limitations of the old components.

Version 128: 31 August 2009

Summary:

Support for Data Entitlements and a major protocol improvement.

Enhancements

New methods for the Message Queue configuration

A set of accessor methods has been added to `MkvMQ` and `MkvMQSubscribe` classes to allow retrieval of configuration objects.

For more information, refer to the *Java API Reference Guide*.

New features and changes

Support for Data Entitlements

The Java API now supports the ION Data Entitlements service.

Data Entitlements allow the centralized control of the data flowing on the ION Platform, and the centralized restriction of access to the data. Access to data means the ability to subscribe any static or dynamic information exposed by any components on the platform.

In this model, access to data available on the platform from a subscriber is subject to the Data Entitlements service. To access data, the subscriber application will require permission from this service.

The Java API now implements the subscription logic subject to entitlements transparently from the application, executing the required queries on the Entitlements Server database when required. Therefore, the usual Publish/Subscribe pattern is maintained as the standard paradigm that applications should use to subscribe to data on the platform.

The application logic should take into account the possibility that a subscription could asynchronously fail as a result of the entitlements checks.

For more information, refer to the *Entitlements Server: User Guide*.

Support for messages larger than 32 KB

The Java API now supports the exchange of messages larger than 32KB, provided the peer component can handle that size.

To facilitate this, a new configuration property, DTBLOCKSIZE, has been added. This property specifies the maximum size of messages that can be sent by the application. The allowed range is from 32KB to 1GB, and the default is 256KB.

For more information about the DTBLOCKSIZE property, refer to the configuration database on the ION Tracker website.

Messages that exceed the DTBLOCKSIZE size are trimmed on a per field basis to fit the maximum value specified.

Function call invocations and responses and/or types that exceed the size are aborted.

Fixes

Fix to prevent the loss of CS types at router registration time (Issue ID 00087706):

The API would sometimes try to register twice to the CS because the first attempt would not complete within the expected timeframe (1 second).

In some cases, the first registration attempt would succeed and the CS would reject the second because the API was already registered.

When managing the failed connection, the API would clear the description of the types received from the CS of the established connection.

The API was consequently unable to supply updates to records using centralized CS types.

Version 127: 24 April 2009

Summary

Support for Message Queues.

New features and changes

Support for message queues

A *message queue* is a mechanism that provides point-to-point communication between components. Delivery of messages is guaranteed through a single named object which is published on the platform and only utilized by the pair of communicating components (queue publisher and subscriber).

Message queues are suitable for exchanging:

- Relatively static data, especially large sets of objects, such as instrument reference data, trades, and historical information.
- Custom sets of data, requested through custom queries, such as data meeting custom criteria specified by the subscriber.

In these cases, message queues are much more flexible and efficient than publishing/subscribing through separate records.

Setting up a message queue entails creating a configuration object and passing it to an `MkvMQManager` instance to obtain an `MkvMQ` object instance. Records can then be written to the message queue.

The publisher can dispose of a message queue object at any time. Calling its close method with a timeout will allow a subscriber to consume the remaining queued records if it manages to do so before the timer elapses.

Support for Entitlement Service (by-feature entitlements)

By-feature entitlements can be defined and utilized by components to prevent unauthorized access to available features.

The Java API now supports the entitlement service facility by exporting the new `MkvEntitlementManager` class, which provides the methods to manage entitlements.

Features represent the operations applicable to services that an application wants to make subject to user entitlements. For example, the ability to create trades on a book or change a pricing configuration on a security.

For more information, refer to the ION Java API documentation and the *Entitlements Server User Guide and Reference*.

Support for multiple return values in function results

This feature, already available in the C-API, is now also provided by the Java API and supported by the routing infrastructure. A new public constructor has been added to `MkvFunction` class to provide the new feature.

- **Subscriber side:**
  The function call interface is unchanged. Results are provided as additional internal fields of the `MkvSupply` object, which was already returned in the previous version but with a single value.
- **Publisher side:**
  The constructor of the `MkvFunction` class has been extended to declare multiple return values.

Improved memory statistics:

The memory utilization statistics reported by the Java API have been improved. In particular:

- `MemSize` is now calculated as the JVM heap used by the application instead of the total JVM heap. • `MemPct` is now calculated as the following ratio:

  ```
  (MemSize / <Xmx value>)
  ```

  That is, the percentage of the memory allocated in the heap with respect to the maximum memory allocable by the JVM. This value should never approach 100, or the application may fail due to Memory Allocation problems.

New E2E calculation algorithm

The End-to-End (E2E) delay calculation algorithm has been improved.

The new algorithm provides more accurate time statistics when the user-provided timestamps are not sampled from the local system clock.

Support for multiple time resolutions

Support for multiple time resolutions, already available in the C-API, is now provided also by the Java API. The following time precisions can be set:

- Hundredth of seconds (1 / 100)
- Milliseconds (1 / 1000)
- Tenths of millisecond (1 / 10000)
- Microseconds (1 / 1000000)

The precision can be set using the `mkv.usemsecs` property in the `mkv.jinit` file. In particular, this setting affects the precision of:

- Timestamps used in the Round Trip Delay (RTD) probes, which are calculated according to the precision specified in the property
- Log file timestamps
- Supply object timestamps

For more information, refer to the Configuration Database on the ION Tracker website.

Encrypted password from properties file

The Java API now supports encrypted passwords in the `mkvpwd` variable by using the same format of the C API.

- Encrypted passwords are identified by a ':' character prefix.
- Encrypted passwords are generated by using the mkvpwdtool.
- These passwords can now be correctly loaded from the Java API through the standard property access methods.

Interface to retrieve platform users and groups

Two new classes are now available, `MkvUser` and `MkvGroup`, to allow users to access information about platform users and their group membership. For more information, refer to the Java API documentation.

Fixes

The publish filtering policy no longer filters out types

Type publishes are now always notified regardless of the value of the following configuration variables:

- `mkv.publish_include_originator`
- `mkv.publish_exclude_originator`
- `mkv.publish_include`
- `mkv.publish_exclude`

Deadlock in case of concurrent subscription and unsubscription of the same chain

The deadlock was caused by a double lock when clearing the cache.

Race condition detected during type re-publish

A type could not be republished if a new record associated with a type being unpublished was received while the Java API was unpublishing records associated with that type.

Fix for potential delays when sending large packets

The delay occurred when the Java API pushed more than 32 KB of data on the outgoing queue in the interval that elapsed between a thread activation request and the actual activation, and no further messages were added to the queue.

Stale update on re-subscribing a field (Issue ID: 00108789)

When subscribing a field which had been previously unsubscribed by the application, a stale update could be immediately delivered on that field.

This happened in rare circumstances, resulting in a race condition between the field unsubscription and the delivery of an update for that field.

Version 125p7: 31 October 2008

Summary

Memory leaks on `Mkv.stop` call.

Fixes

Missing Records when using PUBLISH_INCLUDE option

Filters activated by `PUBLISH_INCLUDE` were affected by a concurrency bug that could cause the API to filter out objects that were not expected to be filtered.

Memory leaks occur when calling the Mkv.stop (Issue ID: 00077821)

After an `Mkv.stop` call, a reference to an `Mkv` object was retained causing memory leaks.

Version 125p6: 09 September 2008

Summary

Fix for NullPointer exception.

Fixes

NullPointerException thrown when registering to the platform and trying to read properties

`MkvProperties.getProperty()` could throw a `NullPointerException` if it was called providing an unregistered property name.

The exception was thrown only if `MkvProperties.getProperty()` was invoked after the component had registered with the platform.

Version 125p5: 24 July 2008

Summary

Fix on Persistent Subscription.

Fixes

Fix on persistent subscription (Issue ID: 00071817)

In some circumstances, a publisher could deliver supplies to a subscriber even if the fields the subscriber requested did not change.

Version 125p4: 04 March 2008

Summary

Fix on External Authentication.

New features and changes

Added the MkvObject.checkIdentifier and MkvPlatform.checkIdentifier methods

These two new methods can be used to check whether the given identifier is acceptable as a publish identifier or as a component identifier.

Added MkvQoS.setLoadInit method

This method can be used to disable the automatic loading of the `mkv.jinit` file.

Fixes

Ineffective extension of subscriptions (Issue ID: 00069518)

It could happen that the extension of the set of subscribed fields was not managed by the API. As a result, the newly subscribed fields were never received by the subscriber application.

Fix on external authentication (Issue ID: 00065912)

The external authentication mechanism did not work when used on Daemon-based platforms.

An error message is always reported if mkv.jinit file cannot be found, even if it may not be needed

Now the user can programmatically disable the loading of the `mkv.jinit` file.

Version 125p1: 31 July 2007

Summary

Changes in the Saving Policy for MkvDB files and Enable Tracing of MkvDB Errors.

New features and changes

Enable tracing of errors when handling files under MKvDB

The new `mkv.dblogerr` configuration variable has been added.

When this option is enabled (set to `1`), the Java API traces errors encountered while handling files stored in the MkvDB directory.

- Values: 0, 1
- Default: 0

For more information, refer to the Configuration Database on the ION Tracker website.

Saving policy changes for connection files:

A new policy has been designed to prevent files corrupting while saving MKvDB files on the disk.

Temporary files are used when updating MkvDB files. The temporary files are then renamed only when the update has been successfully executed.

In case of an error, the original file will not be affected, preserving its integrity.

Version 125: 20 June 2007

Summary

Support for External Authentication.

New Features and Change

Support for external authentication system

Application linking against this API will support the external authentication system.

Enabling components to establish a full duplex connection to the CS

A New MKVSYSCOMPONENT configuration variable has been added.

When this setting is enabled (set to `1`), the component establishes a full duplex connection to the CS.

As a result, the CS will be able to receive publication objects from the component and fully use the pub / sub and req / reply logic available within the PSH protocol.

- Values: 0, 1
- Default: 0

For more information, refer to the Configuration Database on the ION Tracker website.

Fixes

MkvFunctionCallEvent.setErr with 0 code

`MkvFunctionCallEvent.setErr` no longer accepts the `0` code because it is used to mark a reply with no error.