Ignition 8.x Compatible Release Notes

✓ Sign up here to subscribe to our mailing list for software update notifications

⚠ Click here for Ignition Version Compatibility

! Whilst we run rigorous automated testing for each of our modules, we always recommend that you thoroughly test new module versions in a test environment before releasing to production

! **When installing Cirrus Link modules in Ignition, all module versions must match. If they are not, class loading issues may result in system instability. Always make sure all Cirrus Link module versions match in an Ignition Gateway.**

Cirrus Link Modules for Ignition 8.x

Expand/Collapse All to allow Search

- Added support for auto-detection and loading of new TLS certificates when they are loaded into Ignition
- Added support for 'Unique LWT Topics' to reject connections from clients with duplicate MQTT Will Message Topics

- No fixes

- Modified MQTT Engine to disconnect/reconnect when detecting an invalid STATE message to force timestamp update to current time
- Added support for metricTimestampValidationSkew via a CustomProperty in MQTT Engine
- Modified Engine server set creation to automatically add the new set to all namespaces
- Added support for configurable alarm 'display path' at MQTT Engine which propagating alarms from Transmission to Engine
- Modified the Engine and Transmission to support DataSets, Documents, and Array types as Strings on the wire as an option to remain Sparkplug compliant
- Added support for clearing alarm events already cleared at the Edge and missed by MQTT Engine
- Improved logging around null dataset handling
- Added support for 'auto-reconciliation' of alarms between Transmission and Engine
- Added support for auto-removal of Namespace Server Set references when deleting Server Sets

- Fixed potential synchronization issue when removing subscriptions on disconnect
- Fixed updates to the PHID info tags in Engine to not update improperly on incoming older STATE messages
- Fixed handling of Date and String datatypes in alarm propagation
- Fixed alarm preservation across Engine disable/enable and alarm SQL inserts
- Fixed alarm handling to prevent duplication if both the Default and UNS tag tree views are enabled
- Fixed handling of historical alarms as regular metrics which was updating tag values incorrectly
- Fixed potential NPE in handling null bound property values
- Fixed potential edge cases in Dataset handling

- Modified global and edge node level refresh logic to better handle refresh complete
- Made CMD writes non-blocking to improve write performance
- Added support for propagation of all alarm properties when publishing alarms
- Modified Transmission to support DataSets, Documents, and Array types as Strings on the wire as an option to remain Sparkplug compliant
- Added support for 'auto-reconciliation' of alarms between Transmission and Engine
- Added support for manual alarm reconciliation via a Transmission control tag
- Removed code that built up UDT definitions even when they were not configured to be published
- Modified default history flush quantity to 40k

- Fixed rolling buffer store and forward support in Transmission
- Fixed handling of Date and String datatypes in alarm propagation
- Fixed potential concurrency issues in BIRTH building
- Fixed issue when flushing in-order and 'writing directly to the historian' in Engine could result in old/invalid tag values
- Fixed potential NPE during BIRTH building reconciliation
- Fixed potential edge cases in Dataset handling
- Fixed Transmitter config to not allow a Device ID to be configured when an Edge Node ID is not configured

- No new features

- No fixes

- Modified global and edge node level refresh logic to better handle refresh complete
- Added support optionally to include Node/Device Info and Control metrics
- Removed code that built up UDT definitions even when they were not configured to be pushed
- Fixed Agent config to not allow a Device ID to be configured when an Edge Node ID is not configured

- Fixed potential concurrency issues in BIRTH building
- Fixed potential NPE during BIRTH building reconciliation

- Modified global and edge node level refresh logic to better handle refresh complete
- Added ISO-8859-1 encoding support
- Added support optionally to include Node/Device Info and Control metrics
- Removed code that built up UDT definitions even when they were not configured to be pushed
- Fixed Agent config to not allow a Device ID to be configured when an Edge Node ID is not configured

- Fixed potential concurrency issues in BIRTH building
- Fixed potential NPE during BIRTH building reconciliation

- Modified global and edge node level refresh logic to better handle refresh complete
- Added support optionally to include Node/Device Info and Control metrics
- Removed code that built up UDT definitions even when they were not configured to be pushed
- Fixed Agent config to not allow a Device ID to be configured when an Edge Node ID is not configured

- Fixed potential concurrency issues in BIRTH building
- Fixed potential NPE during BIRTH building reconciliation

- No new features

- Fixed potential NPE in the getRegisterSets method of the PollGroupManager class
- Fixed a bug when updated global definitions name is not properly applied to device connections
- Fixed potential NPE in Meter Config record handling

- Improved logging in the initializeOpCodeTables method of the RocDevice class

- Fixed a bug when updated global definitions name is not properly applied to device connections

- No new features

- No fixes

- No new features

- No fixes

- No new features

- Removed invalid update of online/offline timestamp update for Elecsys namespace that resulted in noisy log messages

- No new features

- Fixed issue with threading that resulted in significant performance decrease

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- Upticked Google Protobuf version to v3.25.5

- No fixes

- Upticked Google GSON version to v2.8.9
- Upticked Google Protobuf version to v3.25.5
- Added support to store alarm events in the alarm journal
- Added Sparkplug IDs to the source and display path of alarms at MQTT Engine
- Added support to treat 'all numbers as floats' in Engine JSON custom namespace handling

- Fixed potential deadlock in the alarm DB
- Fixed custom namespace issue that could result in processing topics not intended for that namespace
- Fixed initialization order in MQTT Engine to ensure TLS certificates get loaded on startup
- Fixed support for Array, Dataset, and Document tag properties
- Fixed Elecsys namespace $sys commands

- Upticked Google Protobuf version to v3.25.5
- Added support for 'randomizing server connections'
- Added warning message if a conflicting UDT definition is found and would've been included in the NBIRTH

- Fixed potential deadlock in the disk-backed history store
- Added code to remove RedundancyStateListeners to prevent erroneous error messages
- Fixed birth building issues that could result in erroneous tags being published
- Fixed support for Array, Dataset, and Document tag properties
- Fixed issue that could result in Transmission ceasing to publish data after a Rebirth request
- Fixed potential NPE when trying to detect tag enabled status on startup
- Fixed issue with Dataset datatype coercion
- Fixed UNS Transmitters to include Sparkplug IDs for UDT member tags
- Fixed bound value string handling for numeric bound UDT parameters
- Fixed tag change event reconciliation logic when publishing cached births
- Fixed issue that could result in DATA messages being published out of order when under heavy load

- No new features

- No fixes

- Upticked Google Protobuf version to v3.25.5
- Upticked AWS SDK version to v2.28.17
- Added support for 'hybrid push'
- Added warning message if a conflicting UDT definition is found and would've been included in the NBIRTH

- Added code to remove RedundancyStateListeners to prevent erroneous error messages
- Fixed birth building issues that could result in erroneous tags being pushed
- Fixed potential NPE when trying to detect tag enabled status on startup
- Fixed issue with Dataset datatype coercion
- Fixed bound value string handling for numeric bound UDT parameters
- Fixed tag change event reconciliation logic when pushing cached births

- Upticked Google Protobuf version to v3.25.5
- Added support for 'hybrid push'
- Added warning message if a conflicting UDT definition is found and would've been included in the NBIRTH

- Added code to remove RedundancyStateListeners to prevent erroneous error messages
- Fixed birth building issues that could result in erroneous tags being pushed
- Fixed potential NPE when trying to detect tag enabled status on startup
- Fixed issue with Dataset datatype coercion
- Fixed bound value string handling for numeric bound UDT parameters
- Fixed tag change event reconciliation logic when pushing cached births
- Fixed potential deadlock during disconnect/reconnect events

- Upticked Google Protobuf version to v3.25.5
- Added support for 'hybrid push'
- Added warning message if a conflicting UDT definition is found and would've been included in the NBIRTH

- Added code to remove RedundancyStateListeners to prevent erroneous error messages
- Fixed birth building issues that could result in erroneous tags being pushed
- Fixed potential NPE when trying to detect tag enabled status on startup
- Fixed issue with Dataset datatype coercion
- Fixed bound value string handling for numeric bound UDT parameters
- Fixed tag change event reconciliation logic when pushing cached births

- Upticked Quartz version to v2.3.2

- Added code to remove RedundancyStateListeners to prevent erroneous error messages
- Fixed potential NPEs when processing some types of Totalflow applications

- Upticked Quartz version to v2.3.2

- Added code to remove RedundancyStateListeners to prevent erroneous error messages

- No new features

- No fixes

- Added retry logic for listeners that failed to initialize

- No fixes

- Added 'Not Licensed' state to Server Connection Status field in the Web UI
- General logging improvements
- Added cleanup mechanism to delete Persistent Records associated with a deleted NamespaceRecord
- Added support for null/empty 'replacement string' in custom namespace conversion rules
- Added support to automatically populate new Custom Namespace Server Set bindings list with all existing Server Sets
- Added support for null JSON payload values if the tag already exists
- Added support for Edge side 'Node Control' tags in DATA messages

- Fixed issue that would keep removed legacy state client subscriptions after they were deleted
- Fixes to ensure all subscriptions are necessary
- Fixed decoding support of Array, DataSet, and Document property types
- Improved handling for historical metric DataTypeClass discovery to prevent NPEs

- Added 'Not Licensed' state to Server Connection Status field in the Web UI
- Optimized births to omit unused UDT definitions in the payload
- Added support for the 'Enabled' property to be a bound parameter
- Changed history store query timeout and minimum value to prevent query timeouts

- Fixed potential deadlock in connection logic
- Fixed potential ClassCastException during birth building
- Fixed encoding support of Array, DataSet, and Document property types

- Added documentation tag property for MQTT Recorder tags

- No fixes

- Added documentation tag property for AWS Injector tags
- Optimized births to omit unused UDT definitions in the payload
- Added support for the 'Enabled' property to be a bound parameter
- Changed history store query timeout and minimum value to prevent query timeouts

- Fixed potential deadlock in connection logic
- Fixed potential ClassCastException during birth building
- Fixed encoding support of Array, DataSet, and Document property types

- Added documentation tag property for Azure Injector tags
- Optimized births to omit unused UDT definitions in the payload
- Added support for Azure cert based auth for Event Hub connections
- Added support for the 'Enabled' property to be a bound parameter
- Changed history store query timeout and minimum value to prevent query timeouts

- Fixed potential deadlock in connection logic
- Fixed potential ClassCastException during birth building
- Fixed encoding support of Array, DataSet, and Document property types

- Added documentation tag property for Google Cloud Injector tags
- Optimized births to omit unused UDT definitions in the payload
- Added support for the 'Enabled' property to be a bound parameter
- Changed history store query timeout and minimum value to prevent query timeouts

- Fixed potential deadlock in connection logic
- Fixed potential ClassCastException during birth building
- Fixed encoding support of Array, DataSet, and Document property types

- Added validator to prevent submission of a global definition w/o specifying a file

- Fixed potential NPE around wildcard characters while parsing INI files with @ROWLOOP

- No new features

- No fixes

- No new features

- No fixes

- Added documentation property to info tags to describe each tag

- No fixes

- Added documentation property to info tags to describe each tag
- Made alarm persistence location configurable
- Added documentation property to info tags to describe each tag
- Added the 'ActiveSubscription' Engine Info tag

- Modified python script APIs to not return private keys in plain text
- Logging improvements
- Added support for Alarm ACKs even when command blocking is enabled
- Moved default H2 DB location for Windows from 'data' to 'user-lib'

- Fixed alarm handling of 'ack time', 'clear time', and 'display path'
- Fixed potential erroneous tag states when both UNS and default Sparkplug tags were enabled
- Fixed issue that could result in incorrect filters when filtering on a per server basis
- Fixed alarm ACKs when using Engine on Windows OS
- Fixed logging and feedback when an alarm is ACKed and the associated Edge Node is offline

- Added support for UNS Transmitters
- Added support for alarm 'notes'
- Made disk-backed history store persistence location configurable
- Added documentation property to info tags to describe each tag
- Modified python script APIs to not return private keys in plain text
- Moved default H2 DB location for Windows from 'data' to 'user-lib'
- Improved Exception handling when building BIRTHs to denote datatype mismatches
- Added RedundnacyTransitionActiveDelay config option to Transmission to allow redundant gateway to stop before the active one starts

- Fixed alarm handling of 'ack time', 'clear time', and 'display path'
- Fixed a deadlock that could occur on rebirth requests or in certain startup scenarios
- Fixed SecurityContext handling when not validating writes against a supplied security context
- Fixed issue that could result in in-order history flushing not being honored when a Transmission server config is updated
- Fixed NPE when alarm config contained unknown property
- Fixed situations that could result in an 'unregistered' alarm
- Fixed updating/unregistering a disabled Transmitter

- No new features

- No fixes

- Made disk-backed history store persistence location configurable
- Added documentation property to info tags to describe each tag
- Moved default H2 DB location for Windows from 'data' to 'user-lib'
- Improved Exception handling when building BIRTHs to denote datatype mismatches
- Added support for 'SendAllProps' to send all default tag properties in a BIRTH

- Fixed dynamic handling of services after creating, updating, or deleting agents
- Fixed issue that could result in startup or shutdown of wrong Kinesis endpoint on config changes
- Modified JSON Metrics to always include a value even if it is null

- Made disk-backed history store persistence location configurable
- Added documentation property to info tags to describe each tag
- Modified python script APIs to not return private keys in plain text
- Moved default H2 DB location for Windows from 'data' to 'user-lib'
- Improved Exception handling when building BIRTHs to denote datatype mismatches
- Added support for 'SendAllProps' to send all default tag properties in a BIRTH

- Fixed dynamic handling of services after creating, updating, or deleting agents
- Modified JSON Metrics to always include a value even if it is null

- Made disk-backed history store persistence location configurable
- Added documentation property to info tags to describe each tag
- Modified python script APIs to not return private keys in plain text
- Moved default H2 DB location for Windows from 'data' to 'user-lib'
- Improved Exception handling when building BIRTHs to denote datatype mismatches
- Added support for 'SendAllProps' to send all default tag properties in a BIRTH

- Fixed dynamic handling of services after creating, updating, or deleting agents
- Modified JSON Metrics to always include a value even if it is null

- Added initial support for AAR Alarms

- Fixed an issue with incorrect EFM device connection status when 2-hours trial expires

- No new features

- Fixed potential migration issue with global TLP handling when upgrading from Ignition 7 to 8
- Fixed an issue with incorrect EFM device connection status when 2-hours trial expires

- No new features

- No fixes

- No new features

- No fixes

- No new features

- Fixed Alarm handling to support clearing of alarms before acknowledgement

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- Fixed issue that could cause Distributor to disconnect clients due to not receiving a PINGRESP even when data in still being received
- Fixed socket close handling when no data is sent to support health checks

- Added support for Alarm events via MQTT with Acknowledgments
- Added ‘UNS Tag Tree’ support
- Added ‘Delete UNS’ control tags to better support Edge Node deletion in UNS namespaces
- Added info tags in MQTT Engine to better track the inbound message queue health
- Added configuration option for inbound message thread pool sizing
- Removed ‘URL validation’ is MQTT Engine that prevented duplicate IP/hostname definitions
- Added support for Python system.cirruslink.engine.publish to take either a Server Name or a Server Set name
- Added support for a configurable rebirth debounce delay instead of a hardcoded 5s

- Improved thread/queue handling for inbound messages to prevent ‘rebirth storms’
- Fixed issue where MQTT Engine ‘overrides’ UDT instance member properties if the user has manually modified the Engine side UDT definition member properties
- Fixed Engine server set migration to create a server set per server definition from Engine prior to adding server set support
- Fixed issue that could cause MQTT Engine to not properly publish rebirth requests in some cases

- Added support for Alarm events via MQTT with Acknowledgments
- Modified RPC clients to only be one per server set
- Added a configuration option to delay birth fetching to allow consumption of tag change events that may occur with timestamps before ‘now’ and birth fetching
- Added support for handling of multiple metrics in NCMD and DCMDs
- Moved payload logging from TransmissionClientManager to TransmissionClient

- Fixed issue where MQTT Engine ‘overrides’ UDT instance member properties if the user has manually modified the Engine side UDT definition member properties
- Fixed birth creation to reconcile the birth with tag change events during the birth building process
- Fixed issue that could result in duplicate or incorrect generated aliases
- Fixed issue that could cause Transmission to erroneously flush history when in-order after failing to publish a birth sequence
- Fixed timestamp handling of Array, Document, and Dataset types now that Ignition resolved timestamp handling in Ignition 8.1.43
- Added synchronization around RPC client startup/shutdown to prevent potential thread leak
- Fixed escaping of table names in the disk-backed history store to support ‘-’ chars in Sparkplug IDs
- Added/fixed support for null dataset values

- No new features

- No fixes

- Added a configuration option to delay birth fetching to allow consumption of tag change events that may occur with timestamps before ‘now’ and birth fetching

- Fixed birth creation to reconcile the birth with tag change events during the birth building process
- Fixed timestamp handling of Array, Document, and Dataset types now that Ignition resolved timestamp handling in Ignition 8.1.43

- Fixed escaping of table names in the disk-backed history store to support ‘-’ chars in Sparkplug IDs
- Added/fixed support for null dataset values

- Added a configuration option to delay birth fetching to allow consumption of tag change events that may occur with timestamps before ‘now’ and birth fetching

- Fixed birth creation to reconcile the birth with tag change events during the birth building process
- Fixed timestamp handling of Array, Document, and Dataset types now that Ignition resolved timestamp handling in Ignition 8.1.43
- Fixed escaping of table names in the disk-backed history store to support ‘-’ chars in Sparkplug IDs
- Added/fixed support for null dataset values

- Added a configuration option to delay birth fetching to allow consumption of tag change events that may occur with timestamps before ‘now’ and birth fetching

- Fixed birth creation to reconcile the birth with tag change events during the birth building process
- Fixed timestamp handling of Array, Document, and Dataset types now that Ignition resolved timestamp handling in Ignition 8.1.43
- Fixed escaping of table names in the disk-backed history store to support ‘-’ chars in Sparkplug IDs
- Added/fixed support for null dataset values

- No new features

- Improved ABB CSV parsing to prevent edge case Exceptions
- Fixed migration bug around CSV file handling
- Potential problem when parsing @ROWLOOP @REPLACE with increment above 1

- No new features

- No fixes

- No new features

- No fixes

- No new features

- Fixed connection issues with clients with no specified client ID
- Fixed possible minor delay in processing QoS 2 messages
- Fixed packet identifier warnings in log

- Added support for 'server sets' to MQTT Engine to better support redundancy and more complex architectures
- Added URL validation to server records
- Added support for a Sparkplug 'device dataset' in Engine info tags to show all Sparkplug Devices
- Cleanup/refinement of various info tags
- Added config UI validation to Topic Token Filters
- Added support for AWT Colors in DataSets
- Added support for Array, DataSets, and Document tag properties

- Fixed handling of the Primary Host State info tag when Engine is disabled and config changes
- Fixed client connections to publish an LWT before unsubscribing from Sparkplug Topics
- Fixed NPE in edge cases around Rebirth requests
- Various fixes to UDT collision detection
- Fixed STATE correction handling in redundant and multi-server environments
- Fixed potential thread leak on shutdown of MQTT clients
- Fixed the DEATH message handling so tags with null values are properly 'staled'

- Added URL validation to server records
- Added a thread pool to hand off history store writes to keep data flowing smoothly out of Transmission
- Added support for 'maximum history store size' to prevent history from overrunning memory or disk
- Added support for AWT Colors in DataSets
- Added a config UI validator to ensure PHID is set when file 'auto-publish' is enabled as it is required
- Added support for Array, DataSets, and Document tag properties
- Made the H2 TCP port configurable
- General logging cleanup
- Increated Transmission RPC clients to support 500 in-flight MQTT messages
- General performance improvements to disk-backed history storage

- Fixed History Store category headings in the config UI
- Fixed handling of restarted tags to ensure they properly return to a good quality
- Fixed issue that prevented proper publishing when child UDTs are deleted from a parent UDT and 'convert UDTs' is enabled
- Moved history store on disk to the 'data' directory to better support redundancy and gateway backups
- Fixed permissions issue that prevented Transmission from browsing tags when read permissions had been modified on the tag provider
- Various fixes to UDT collision detection
- Fixed startup race condition that could result in invalid initial values
- Various fixes to Transmission info tags
- Fixed potential NPEs when fetching existing tag paths
- Fixed handling of numeric parameter values in UDTs

- No new features

- No fixes

- Added support for EC2 Instance IAM Roles as an alternative to AWS access/secret keys
- Added support for 'maximum history store size' to prevent history from overrunning memory or disk
- Added support for AWT Colors in DataSets
- Added support for Array, DataSets, and Document tag properties
- Made the H2 TCP port configurable
- General logging cleanup
- General performance improvements to disk-backed history storage

- Fixed handling of restarted tags to ensure they properly return to a good quality
- Moved history store on disk to the 'data' directory to better support redundancy and gateway backups
- Fixed permissions issue that prevented Injectors from browsing tags when read permissions had been modified on the tag provider
- Fixed startup race condition that could result in invalid initial values
- Fixed potential NPEs when fetching existing tag paths
- Fixed handling of numeric parameter values in UDTs
- Fixed dynamic configuration UI handling in Kinesis configuration
- Fixed support for Kinesis Firehose configurations

- Added support for 'maximum history store size' to prevent history from overrunning memory or disk
- Added support for AWT Colors in DataSets
- Added support for Array, DataSets, and Document tag properties
- Made the H2 TCP port configurable
- General logging cleanup
- General performance improvements to disk-backed history storage

- Fixed handling of restarted tags to ensure they properly return to a good quality
- Moved history store on disk to the 'data' directory to better support redundancy and gateway backups
- Fixed permissions issue that prevented Injectors from browsing tags when read permissions had been modified on the tag provider
- Fixed startup race condition that could result in invalid initial values
- Fixed potential NPEs when fetching existing tag paths
- Fixed handling of numeric parameter values in UDTs
- Fixed potential NPE when using certificate based auth and no CA cert is defined

- Added support for 'maximum history store size' to prevent history from overrunning memory or disk
- Added support for AWT Colors in DataSets
- Added support for Array, DataSets, and Document tag properties
- Made the H2 TCP port configurable
- General logging cleanup
- General performance improvements to disk-backed history storage

- Fixed handling of restarted tags to ensure they properly return to a good quality
- Moved history store on disk to the 'data' directory to better support redundancy and gateway backups
- Fixed permissions issue that prevented Injectors from browsing tags when read permissions had been modified on the tag provider
- Fixed startup race condition that could result in invalid initial values
- Fixed potential NPEs when fetching existing tag paths
- Fixed handling of numeric parameter values in UDTs

- Added 'Flow Factor', 'Flow Time', and 'Total Volume' column support

- Fixes to migration code for Totalflow applications
- Fix to restart polling when auto-discovery completes and ON_DEMAND is being used
- Fixed flowtime calculation as it was improperly being multiplied by 60

- Added 'Flow Factor', 'Flow Time', and 'Total Volume' column support

- No fixes

- No new features

- No fixes

- Added custom property support for 'maxOutboundQueueSize'

- No fixes

- MQTT Engine now excludes empty property sets on the wire which could result in problems for some Edge Node implementations
- Modified MQTT Engine to only perform UDT collision detection for MQTT Transmission Sparkplug Edge Nodes
- Improved logging to not log at info level on ignored tag collisions when a duplicate tag change event is being ignored and is identical to the original
- Added support for excluding datatypes from CMD messages per the Sparkplug spec
- Added support for missing datatypes in DATA messages per the Sparkplug spec
- Added support for 'Binary Data' datatype
- Added Engine side write support for ByteArray, DateTimeArray, and StringArray datatypes

- Fixed potential erroneous UDT collision warnings in MQTT Engine
- Fixed potential NPE in UDT collision detection for templates with no parameters
- Fixed timing issue that could result in Transmission attempting to read tags before the tag provider is started which could result in failure to initialize on startup

- Made 'hashing algorithm' names consistent across all modules
- Increased the default query timeout for disk-backed history stores to improve reliability on underpowered systems
- Added support for excluding datatypes from DATA messages per the Sparkplug spec

- Added support for missing datatypes in CMD messages per the Sparkplug spec
- Added a dedup filter to prevent some edge cases where BIRTH and subsequent DATA messages could include identical tag change events
- Added support for 'Binary Data' datatype
- Set defaults for history store configurations

- Fixed multiple edge cases where duplicate tag change events could be sent when they shouldn't be
- Fixed edge case that could result in some metrics failing to get aliases assigned when aliasing is enabled
- Fixed support for ByteArray datatypes
- Fixed potential NPEs when flushing history in-order
- Fixed issue that could result in live data being mixed in with history during in-order history flush
- Added Paho bug workaround to minimize leaked Paho threads
- Fixed dataset deserialization when in use in history stores which resulted in zero values in datasets at MQTT Engine
- Fixed issue that could result in missing aliases when using aliasing with history stores
- Removed issue that could result in Ignition shutdown when using disk-backed history store
- Fixed issue that required a CA cert when using client side cert auth even if using a 'real' CA

- Made 'hashing algorithm' names consistent across all modules

- No fixes

- Increased the default query timeout for disk-backed history stores to improve reliability on underpowered systems
- Added support for 'Binary Data' datatype

- Fixed multiple edge cases where duplicate tag change events could be sent when they shouldn't be
- Fixed timing issue that could result in Transmission attempting to read tags before the tag provider is started which could result in failure to initialize on startup
- Fixed support for ByteArray datatypes
- Fixed dataset deserialization when in use in history stores which resulted in zero values in datasets
- Added code to ensure a history store is not started if the Injector is disabled
- Removed issue that could result in Ignition shutdown when using disk-backed history store

- Increased the default query timeout for disk-backed history stores to improve reliability on underpowered systems
- Added support for 'Binary Data' datatype

- Fixed multiple edge cases where duplicate tag change events could be sent when they shouldn't be
- Fixed timing issue that could result in Transmission attempting to read tags before the tag provider is started which could result in failure to initialize on startup
- Fixed support for ByteArray datatypes
- Added Paho bug workaround to minimize leaked Paho threads
- Fixed dataset deserialization when in use in history stores which resulted in zero values in datasets
- Added code to ensure a history store is not started if the Injector is disabled
- Fixed potential IoT Hub deadlock
- Removed issue that could result in Ignition shutdown when using disk-backed history store

- Modified default configuration to use Clearblade IoT Core endpoint
- Increased the default query timeout for disk-backed history stores to improve reliability on underpowered systems
- Added support for 'Binary Data' datatype

- Fixed multiple edge cases where duplicate tag change events could be sent when they shouldn't be
- Fixed timing issue that could result in Transmission attempting to read tags before the tag provider is started which could result in failure to initialize on startup
- Fixed support for ByteArray datatypes
- Added Paho bug workaround to minimize leaked Paho threads
- Fixed dataset deserialization when in use in history stores which resulted in zero values in datasets
- Added code to ensure a history store is not started if the Injector is disabled
- Removed issue that could result in Ignition shutdown when using disk-backed history store

- Made 'hashing algorithm' names consistent across all modules

- No fixes

- Made 'hashing algorithm' names consistent across all modules

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- Added 'actor' support to audit records for MQTT Engine to Transmission writes

- Removed invalid check requiring the Sparkplug sequence numbers be zero in an NBIRTH

- Added Transmission RECORD signature fields to the Python scripting API

- Fixed file publishing 'Reset' tag handling

- Fixed issue that could result in Transmission RECORD config changes not taking effect
- Fixed history store and forward migration to new table setup
- Fixed history store and forward pruning logic for certain edge cases
- Added support for '-' in disk-backed store and forward table names and some other special chars

- No new features

- No fixes

- Logging improvements for better diagnostics

- Fixed enable/disable handling via the 'Enabled' Ignition tag
- Fixed IndexOutOfBoundsExceptions on attempts to create endpoint services and agents w/o sets defined
- Fixed sequence number handling in pushed messages
- Fixed Injector failover by ensuring Service startup as well as Agent startup in redundant Ignition environments
- Fixed issue with history store enablement for a disabled service
- Fixed issue that could result in duplicated BIRTH messages
- Fixed 'Messages Stored' count in Info tag
- Fixed potential thread deadlock
- Fixed history store and forward pruning logic for certain edge cases
- Added support for '-' in disk-backed store and forward table names and some other special chars

- Logging improvements for better diagnostics

- Fixed enable/disable handling via the 'Enabled' Ignition tag
- Fixed IndexOutOfBoundsExceptions on attempts to create endpoint services and agents w/o sets defined
- Fixed sequence number handling in pushed messages
- Fixed Injector failover by ensuring Service startup as well as Agent startup in redundant Ignition environments
- Fixed issue with history store enablement for a disabled service
- Fixed issue that could result in duplicated BIRTH messages
- Fixed 'Messages Stored' count in Info tag
- Fixed potential thread deadlock
- Fixed history store and forward pruning logic for certain edge cases
- Added support for '-' in disk-backed store and forward table names and some other special chars

- Logging improvements for better diagnostics

- Fixed enable/disable handling via the 'Enabled' Ignition tag
- Fixed IndexOutOfBoundsExceptions on attempts to create endpoint services and agents w/o sets defined
- Fixed sequence number handling in pushed messages
- Fixed Injector failover by ensuring Service startup as well as Agent startup in redundant Ignition environments
- Fixed issue with history store enablement for a disabled service
- Fixed issue that could result in duplicated BIRTH messages
- Fixed 'Messages Stored' count in Info tag
- Fixed potential thread deadlock
- Fixed history store and forward pruning logic for certain edge cases
- Added support for '-' in disk-backed store and forward table names and some other special chars

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- Improved logging on clearing of retained messages

- Fixed log messages when client connections pre-empt existing connections with same client ID

- Expanded TagPath validation to ensure paths don't end in a slash
- Added 'parentEnabled' and 'sourceTagPath' to list of default filtered properties
- Added JSON validation for the MQTT Engine custom properties config field

- Fixed writing of array values in MQTT Engine tag provider for null and full arrays
- Fixed legacy STATE MQTT client migration when updating from older versions of MQTT Engine

- Optimized BIRTH building to consolidate readTag calls - helps with remote tag provider tags in BIRTH building
- Expanded TagPath validation to ensure paths don't end in a slash
- Added 'parentEnabled' and 'sourceTagPath' to list of default filtered properties
- Major overhaul of disk-backed store and forward system to optimize storage and configuration
- Added digital signature support to Transmission RECORD creation
- Added defaults for rolling history buffer configuration
- Added Transmission history store field validation
- Removed the 'prune quantity' option for rolling history configuration from the web UI

- Fixed issue where Edge Node level expression tags are null on NBIRTH
- Fixed null value handling in RECORDs that would result in default values being set instead of null
- Fixed cache update code that could result in BIRTH messages not publishing the most recent/valid value
- Fixed potential erroneous data type change event when converting tags for UDT instance members
- Fixed issue where timestamps could be set incorrectly if the tag is in a Managed Tag Provider
- Fixed NPE on initialization
- Fixed handling of CMD messages missing Metric datatypes per the Sparkplug spec including Rebirth requests
- Fixed handling of tag properties that are not of a supported datatype

- No new features

- No fixes

- Optimized BIRTH building to consolidate readTag calls - helps with remote tag provider tags in BIRTH building
- Expanded TagPath validation to ensure paths don't end in a slash
- Major overhaul of disk-backed store and forward system to optimize storage and configuration
- General logging improvements

- Fixed issue where Edge Node level expression tags are null on NBIRTH
- Fixed cache update code that could result in BIRTH messages not publishing the most recent/valid value
- Fixed 'enabled' value handling on setting endpoints
- Fixed potential erroneous data type change event when converting tags for UDT instance members
- Fixed NPE on initialization
- Fixed Injector Info tags that could show incorrect data
- Fixed PUSH_TO_ANY handling in Injectors when transitioning on connection status
- Fixed NPE on fresh module installation
- Fixed handling of tag properties that are not of a supported datatype

- Optimized BIRTH building to consolidate readTag calls - helps with remote tag provider tags in BIRTH building
- Expanded TagPath validation to ensure paths don't end in a slash
- Major overhaul of disk-backed store and forward system to optimize storage and configuration
- General logging improvements
- Added configurable SAS token expiration length

- Fixed issue where Edge Node level expression tags are null on NBIRTH
- Fixed cache update code that could result in BIRTH messages not publishing the most recent/valid value
- Fixed 'enabled' value handling on setting endpoints
- Fixed potential erroneous data type change event when converting tags for UDT instance members
- Fixed NPE on initialization
- Fixed Injector Info tags that could show incorrect data
- Fixed PUSH_TO_ANY handling in Injectors when transitioning on connection status
- Fixed NPE on fresh module installation
- Fixed handling of tag properties that are not of a supported datatype

- Optimized BIRTH building to consolidate readTag calls - helps with remote tag provider tags in BIRTH building
- Expanded TagPath validation to ensure paths don't end in a slash
- Major overhaul of disk-backed store and forward system to optimize storage and configuration
- General logging improvements

- Fixed issue where Edge Node level expression tags are null on NBIRTH
- Fixed cache update code that could result in BIRTH messages not publishing the most recent/valid value
- Fixed 'enabled' value handling on setting endpoints
- Fixed potential erroneous data type change event when converting tags for UDT instance members
- Fixed NPE on initialization
- Fixed Injector Info tags that could show incorrect data
- Fixed PUSH_TO_ANY handling in Injectors when transitioning on connection status
- Fixed NPE on fresh module installation
- Fixed handling of tag properties that are not of a supported datatype

- Added Python scripting APIs to copy configs

- Fixed handling of empty CFX publish cron expressions
- Fixed a web UI validation issue 'Publish Files Folder' field

- Added Python scripting APIs to copy configs
- Improved CRC calculation on ROC messages to account for padding

- Fixed handling of empty CFX publish cron expressions
- Fixed a web UI validation issue 'Publish Files Folder' field

- No new features

- No fixes

- Added additional 'Distributor Info' tags

- No Fixes

- No new features

- Fixed legacy client not disconnecting when control tags force offline
- Fixed issue with cloning UDTs when using arrays
- Fixed issue with overrides on UDT Tag properties

- No new features

- Fixed issue where manually entered client ID could cause RPC client to not start
- Fixed BIRTH building of OPC tags to ensure values are set properly when they are members of UDTs

- No new features

- No fixes

- Added Edge Node refresh support
- Improved Agent and Set refresh/restart logic

- Fixed issue where BIRTH message might not get sent on Kinesis Stream enable/disable

- Added Edge Node refresh support

- Fixed IoT Central connections not publishing BIRTH

- Added Edge Node refresh support

- No Fixes

- No new features

- No Fixes

- No new features

- No fixes

- No new features

- No fixes

- Updated the code signing certificate
- Improved logging around PUBLISH and DISCONNECT packet handling

- Improved session tracking and expiring of expired/disconnected sessions
- Fixed CONNACK error code for failed authentication

- Updated the code signing certificate
- Added 'total metric count' and 'metrics per second' info tags
- Logging improvements around timestamp collisions
- Removed Conscript dependency to better support SSL connections on ARM platforms
- Added support for password protected PKCS8 private SSL keys

- Added NPE protection against arriving null tag values
- Fixed regression in String payload handling that prevented tag updates in non-JSON custom namespaces
- Added null protection on connection lost for clients without a specified LWT/Primary Host ID
- Fixed handling of bound parameters in UDTs

- Updated the code signing certificate
- Optimized Birth building to improve performance
- Added info tags for 'metric count' and 'metrics per second'
- Added 'rolling buffer' support to preserve data during keep alive connection lost events
- Expanded on tag event tracing for debug purposes
- Optimized recursive tag configuration lookups on startup
- Removed Conscript dependency to better support SSL connections on ARM platforms
- Added support for password protected PKCS8 private SSL keys

- Fixed the return type for RandomStartupDelay in Transmission Servers createConfig Python scripting API
- Fixed some edge cases around timestamps on metrics during startup
- Fixed regression that previously broke device topic expansion
- Fixed store and order buffer overflows to delete the oldest data first
- Fixed cache updating issue when handling with disabled tags

- Updated the code signing certificate

- No fixes

- Updated the code signing certificate
- Expanded on tag event tracing for debug purposes
- Added disk-backed store and forward support
- Optimized recursive tag configuration lookups on startup
- Added individual Agent enable/disable support and a name setting
- Removed Conscript dependency to better support SSL connections on ARM platforms

- Fixed FlushPeriod return type in the Python scripting API
- Fixed cache updating issue when handling with disabled tags

- Updated the code signing certificate
- Expanded on tag event tracing for debug purposes
- Added disk-backed store and forward support
- Optimized recursive tag configuration lookups on startup
- Added individual Agent enable/disable support and a name setting
- Removed Conscript dependency to better support SSL connections on ARM platforms
- Added support for password protected PKCS8 private SSL keys

- Fixed cache updating issue when handling with disabled tags

- Updated the code signing certificate
- Expanded on tag event tracing for debug purposes
- Added disk-backed store and forward support
- Optimized recursive tag configuration lookups on startup
- Added individual Agent enable/disable support and a name setting
- Removed Conscript dependency to better support SSL connections on ARM platforms
- Added support for password protected PKCS8 private SSL keys

- Fixed cache updating issue when handling with disabled tags

- Updated the code signing certificate

- Fixed deletion and validation of global definitions and their references

- Updated the code signing certificate

- No fixes

- Updated the code signing certificate

- No fixes

- No new features

- Fixed thread management issue when clients continually reconnect with bad credentials

- Added encryption of the security context hashing password
- Added support for DATA metrics missing the MetricDataType because it is known from the BIRTH
- Added support for topic namespace filters to be scoped to specific MQTT Servers
- Added a custom config property to support Edge Node implementations that don't include BIRTH sequence numbers
- Improved various RPC calls for Python scripting consistency
- Added support for detection and logging of UDT collisions at MQTT Engine when they differ
- Logging improvements and error handling improvements on malformed NBIRTH messages
- Added user warnings for custom namespace subscriptions that include spaces
- Added legacy state info tags to MQTT Engine
- Added auditing support for remote writes in MQTT Transmission
- Moved the 'enable primary host state' config option to the server's legacy state client config section
- Logging improvements

- Fixed to ensuring client shutdown always happens when required
- Fixed STATE message handling in multi-server environments
- Fixed issue that could result in topic subscriptions not being reset properly on filter changes
- Improved error handling in Sparkplug BIRTH processing
- Improved error handling for custom namespace string payloads that can't be converted to Ignition tag paths
- Fixed incorrect OFFLINE state values in the MQTT Engine info tags
- Fixed null elements in array handling
- Fixed Elecsys 'State' topic handling
- Fixed tag/folder management for info tags when enabling primary host subscriptions
- Fixed issue that resulted in Custom Namespace String tags not getting updated after initial creation

- Modified to not publish cached BIRTH messages if no historical data was published
- Added encryption of the security context hashing password
- Added user warnings when users set very large history store capacities
- Added support for 'ExcludeFromPayload' to prevent specific UDTs from being published via MQTT
- Added retry logic for example tag creation on initial Transmission installation
- Added auditing support for remote writes
- Added support for publishing alarm events on tags via MQTT

- Modified Transmission to cache historical births at connection loss time rather than when finished flushing
- Fixed STATE message handling in multi-server environments
- Fixed potential deadlock when using the disk-backed history store
- Fixed various RPC calls for Python scripting consistency
- Fixed issue that resulted in old timestamps being included with newly enabled tags
- Fixed NPE when handling online transition
- Fixed issue that could result in device discovery before a 'Refresh' is performed
- Fixed 'Enabled' property change event reporting on UDT member tags
- Added support to force Paho disconnect if Paho still reports 'online' state to ensure proper reconnect

- No new features

- Fixed various RPC calls for Python scripting consistency

- Added support for 'ExcludeFromPayload' to prevent specific UDTs from being published via MQTT

- Fixed 'Enabled' property change event reporting on UDT member tags

- Added support for 'ExcludeFromPayload' to prevent specific UDTs from being published via MQTT

- Fixed various RPC calls for Python scripting consistency
- Fixed 'Enabled' property change event reporting on UDT member tags

- Added support for a custom endpoint server URL which results in support for Clearblade migrations
- Added support for 'ExcludeFromPayload' to prevent specific UDTs from being published via MQTT
- Added validation for mandatory private key upload field

- Fixed 'Enabled' property change event reporting on UDT member tags

- Improved CFX filename conventions

- Added support to ensure application names during auto-discovery are unique
- Fixed potential NPEs when updating meter and periodic config mappings

- Improved CFX filename conventions
- Modified Opcode table support to include the 'L' value in tag names to prevent name overlap

- Improved OpCode table detection logging to show failed TLPs
- Fixed ROC clock sync to work properly when multiple device are syncing clocks
- Fixed potential NPEs when updating meter and periodic config mappings
- Fixed exception handling when losing connection while in the connection establishment period

- No new features

- No Fixes

- Added return values for Python scripting API CRUD operations

- No Fixes

- Added support for Device Rebirth requests
- Disabled non-Sparkplug-B namespaces by default
- Set max subscriptions to 8 per request to better support some third party MQTT Servers
- Added return values for Python scripting API CRUD operations

- Fixed status indication for devices that fail to subscribe to the MQTT Server
- Fixed Elecsys support for '_' chars in messages

- Added return values for Python scripting API CRUD operations

- Added support for Device Rebirth requests
- Fixed issue that could result in missing UDT metrics in a BIRTH
- Fixed erroneous detection of edge node and group folder removal
- Logging improvements on startup/connect
- Fixed issue where EFM records may not be published at startup
- Fixed NDEATH publishing when the Transmitter is disabled

- Added return values for Python scripting API CRUD operations

- No Fixes

- Added 'refresh required' info tag and removed the 'auto-discover' feature
- Added return values for Python scripting API CRUD operations

- Fixed Agent name initialization that could result in NPE at startup
- Fixed handling of enable/disable of endpoints via the control tag
- Fixed NDEATH publishing when the Agent is disabled
- Updated logic to support connects without BIRTH messages when appropriate
- Fixed Agent validation to allow duplicate records

- Added 'refresh required' info tag and removed the 'auto-discover' feature
- Added return values for Python scripting API CRUD operations

- Fixed Agent name initialization that could result in NPE at startup
- Fixed handling of enable/disable of endpoints via the control tag
- Fixed NDEATH publishing when the Agent is disabled
- Updated logic to support connects without BIRTH messages when appropriate
- Fixed Agent validation to allow duplicate records

- Added 'refresh required' info tag and removed the 'auto-discover' feature

- Added return values for Python scripting API CRUD operations

- Fixed Agent name initialization that could result in NPE at startup
- Fixed handling of enable/disable of endpoints via the control tag
- Fixed NDEATH publishing when the Agent is disabled
- Updated logic to support connects without BIRTH messages when appropriate
- Fixed Agent validation to allow duplicate records

- Added support for Meter Configurations
- Added support for CFX file creation
- Improved INI polling status tracking
- Logging improvements
- Added a 'not licensed' status indicator
- Added support for user-selectable INI files
- Added protection of 'double click' events during config that prevented DB access

- Fixed NPE that could result in N/A device connection status
- Fixed issue that could result in 'empty application name' in config menu
- Fixed issue that prevented access to an internal application map during config
- Fixed issue that resulted in failure to obtain meter config when using PERSISTENT connections

- Added support for CFX file creation
- Added support for OpCode Table polling as templates
- Added a 'not licensed' status indicator
- Added validation for CSV file uploads to ensure proper format

- Fixed race condition that resulted in errors when the history poller is stopped

- Logging improvements

- No Fixes

- Replaced core MQTT Server code with clean room implementation

- Fixed MQTT over Websocket handling
- Fixed NPE during anonymous connections
- Fixed memory leak around uses of QoS greater than 0

- Added support for Sparkplug v3.0.0 STATE messages
- Added optional/configurable legacy STATE support for Sparkplug 2.2 clients
- Logging improvements on Threadpools
- Added a precheck on message decoding Sparkplug messages in case of invalid message payloads
- Added support for custom namespace overlap on incoming messages
- Added support for parenthesis in tag paths
- Added native support for 'reordering timeouts' to support Sparkplug v3.0.0

- Fixed potential NPE and ConcurrentModificationException in Primary Host handling
- Fixed support for the 'enabled' tag property
- Fixed Info tag initialization
- Fixed config validation of multiple topics on a single entry
- Fixed UDT handling for disabled UDTs via the 'enabled' tag property
- Fixed Elecsys sys02 sending to correct server in multi-server environments

- Added support for Sparkplug v3.0.0 STATE messages
- Improved File auto-publisher performance
- Increased maximum number of concurrent file publishers
- Fixed possible deadlocks around file, record, history, and live tag publishing
- Added thread names for improved logging in file publishers
- Added logging of time spent on file publish operations
- Added support for null/empty dataset values
- Optimized File publishing by removing sync on tag change events
- Added force sending of pending publish before DDEATH sending
- Improved inline help text for Store and Forward settings
- Added support for disk-backed Sparkplug bdSeq numbers

- Fixed file 'auto-publisher' threads to terminate on shutdown with performance improvements
- Added synchronization to STATE message handlers to prevent timing issues on STATE handling
- Fixes to online/offline handling and manual file publishing
- Fixed 'Publish File Count' metric value
- Fixed reset of 'Publish Files Folder' value when 'Reset' is set true
- Added NPE protection on handling of null tag configs
- Fixed Topic token validation in config operations
- Fixed support for the 'enabled' tag property
- Added NPE protection when publishing DDEATH messages
- Fixed Info tag initialization
- Fixed potential overwrite of 'Offline DateTime' with null value
- Fixed disk backed history store inserts with special characters
- Fixed validation of Group/Edge ID collisions when they are in different Server Sets

- No new features

- No Fixes

- Optimized 'rebirth' publishing to not rescan the tag tree
- Added force sending of pending publish before DDEATH sending
- Added support for disk-backed Sparkplug bdSeq numbers

- Added NPE protection on handling of null tag configs
- Fixed Topic token validation in config operations
- Fixed potential deadlock when pushing payloads under heavy load

- Optimized 'rebirth' publishing to not rescan the tag tree
- Added force sending of pending publish before DDEATH sending
- Added support for disk-backed Sparkplug bdSeq numbers

- Added NPE protection on handling of null tag configs
- Fixed Topic token validation in config operations
- Fixed potential deadlock when pushing payloads under heavy load

- Optimized 'rebirth' publishing to not rescan the tag tree
- Added force sending of pending publish before DDEATH sending
- Added support for disk-backed Sparkplug bdSeq numbers

- Added NPE protection on handling of null tag configs
- Fixed Topic token validation in config operations
- Fixed potential deadlock when pushing payloads under heavy load

- Optimized 'rebirth' publishing to not rescan the tag tree
- Added force sending of pending publish before DDEATH sending
- Added support for disk-backed Sparkplug bdSeq numbers

- Added NPE protection on handling of null tag configs
- Fixed Topic token validation in config operations
- Fixed potential deadlock when pushing payloads under heavy load

- No new features

- No Fixes

- No new features

- No Fixes

- Added support for DateTime Arrays and String Arrays
- Logging improvements/cleanup

- Fixed issue with file transfers of files with invalid characters in the filename
- Fixed issue with BED Edge Node names
- Fixed issue with Tags not getting updated
- Fixed issue with nested property sets

- Added support for DateTime Arrays and String Arrays
- Added validation to Transmitter record creation to ensure duplicate Sparkplug Edge Node descriptors don’t collide

- Fixed issue with history lost on MQTT Transmission info tags
- Fix for newly added device tag folders
- Fix for Error_TypeConversion error
- Fixed issue with UDT name matching

- No new features

- No Fixes

- No new features

- No Fixes

- No new features

- No Fixes

- Added session token auto-refresh

- No Fixes

- No new features

- No Fixes

- No new features

- No Fixes

- No new features

- No Fixes

- No new features

- No Fixes

- No new features

- No Fixes

- Improvements to handling of redundancy synchronization after record updates

- No Fixes

- Added support for the Ignition tag 'enabled' property to be dynamically handled
- Modified MQTT Engine to not update tags if the incoming metric value is older than the current tag value
- Improvements to handling of redundancy synchronization after record updates
- Logging improvements

- Fixed NPE related to logging
- Added rebirth request when Engine receives unknown alias for a metric in a DATA message
- Fixed issue that caused 'writable custom namespace tags' to become unwritable after a module restart
- Fixed handling of deleted Elecsys Edge Nodes
- Added NPE protection around Engine shutdown when Engine itself is disabled via its own config
- Fixed the validator for the 'Root Tag Folder' to allow slashes in the tag folder

- Modified the 'Auto-Connect RPC client' default to false
- Added support for the Ignition tag 'enabled' property to be dynamically handled
- Improvements to handling of redundancy synchronization after record updates
- Logging improvements
- Added support for the new Sparkplug STATE JSON based payloads

- Fixed issue that prevented UDTs with 'overlapping metric names' from being properly handled
- Fixes to synchronization in the RAM based Store and Forward mechanism
- Fixed config keys for the Transmission 'File Records'
- Fixed concurrency issues around BIRTH building in some complex UDT scenarios

- Improvements to handling of redundancy synchronization after record updates

- No Fixes

- Improvements to handling of redundancy synchronization after record updates

- No Fixes

- Modified config descriptions to better match the new Azure IoT Central documentation
- Added in-editor help text for the Azure IoT Central Python configuration API
- Improvements to handling of redundancy synchronization after record updates

- No Fixes

- Improvements to handling of redundancy synchronization after record updates

- No Fixes

- Improvements to handling of redundancy synchronization after record updates

- No Fixes

- Improvements to handling of redundancy synchronization after record updates
- Added support for SUCOR type applications

- Prevent the one-short record poll after reconnect when in persistent mode

- Added dynamic configuration handling for global definition updates for meter and periodic mappings
- Improvements to handling of redundancy synchronization after record updates

- Fixed potential NPE when polling for periodic, minute, and daily history
- Fixed issue that results in occasional 'History Request Failed' errors
- Fixed potential issue with NPEs in ROC request/response handling

- No new features

- No Fixes

- Improvements to handling of redundancy synchronization after record updates

- No Fixes

- No new features

- Fixed Python Scripting API bug around updating Users

- Added Tag Path validation to configuration
- Removed an unnecessary 'back' link in the Custom Namespace String Conversion page
- Added Engine Info tags for tracking the count of re-ordered messages when enabled
- Added a 'force redundancy sync' after the Primary Host ID is changed, enabled, or disabled via control tags
- Added a feature to allow Engine to hold off on publishing the Host STATE message until conditions are met

- Added NPE protection when handling Aliases
- Fixed Elecsys namespace Edge Node ID filtering
- Fixed handling of Custom Namespace JSON tags to prevent unnecessary/improper reconfiguration of tags
- Added NPE protection to aliased metrics when handling tag writes
- Added NPE protection when 're-ordering messages' that could prevent Edge Nodes from showing as online
- Fixed issue with property updating in UDT members that are in member UDTs and folders
- Fixed Elecsys namespace handling of SINT16 data types

- Added a 'Friendly Name' configuration option for File Records
- Added a 'File Scan Rate Time Unit' configuration option for File Records to allow for faster scans
- Added additional info tags for File publishing including 'Files to Auto-Publish', 'Remaining Retries', and 'Reset'
- Improved thread naming for File Publishers to make them more identifiable
- Added Tag Path validation to configuration
- Added support for treating 'Device level UDTs' as devices when converting UDTs
- Improved Birth caching to increase performance
- Optimized caching to reduce overall memory footprint
- Removed excessive calls to 'getTagConfigsAsync' to increase performance

- Fixed retry logic when publishing files
- Added NPE protection when handling Aliases
- Fixed an issue where updating a File Configuration could result in removing or resetting file info tags
- Fixed issue with birth caching logic that produced failures in flushing history when flushing in-order
- Added NPE protection when updating Transmission Info tags
- Fixed issue that could result in DATA messages being published before the BIRTH sequence when flushing history in-order
- Fixed startup/shutdown synchronization that could result in thread leaks
- Fixed issue when flushing in-order that could result in an endless loop with many devices
- Fixes to Edge Node discovery to ensure discovery and status flags update properly
- Fixed issue with Transmission publishing incorrect (newer) metric timestamp in Birth message when using in-order/RAM store and forward
- Added NPE check when shutting down Transmission clients
- Fixed RejectedExecutionException that could occur when deleting tags Transmission was publishing
- Fixed potential double DEATH publishes that could occur on device folder removal

- Added Python scripting API for scripted configuration

- No fixes

- Improved Birth caching to increase performance
- Removed excessive calls to 'getTagConfigsAsync' to increase performance

- Fixed startup/shutdown synchronization that could result in thread leaks
- Fixed issue that could result in the Push Policy reverting back to 'Push to All'

- Improved Birth caching to increase performance
- Removed excessive calls to 'getTagConfigsAsync' to increase performance

- Fixed startup/shutdown synchronization that could result in thread leaks
- Fixed issue that could result in the Push Policy reverting back to 'Push to All'
- Modified scripting API to use 'AzureSetId' instead of 'AzureSet' for Set key names

- Added TCP port number configuration option for connecting to Google IoT Core
- Improved Birth caching to increase performance
- Added Python scripting API for scripted configuration
- Removed excessive calls to 'getTagConfigsAsync' to increase performance
- Added a 'max throughput' configuration option to throttle data flowing to Google IoT Core

- Fixed startup/shutdown synchronization that could result in thread leaks
- Fixed issue that could result in the Push Policy reverting back to 'Push to All'

- Improved Birth caching to increase performance
- Added Python scripting API for scripted configuration
- Removed excessive calls to 'getTagConfigsAsync' to increase performance

- Fixed startup/shutdown synchronization that could result in thread leaks
- Fixed issue that could result in the Push Policy reverting back to 'Push to All'

- Added Tag Path validation to configuration

- Fixed meter type handling during auto-discovery when using applications other than AGA3, SUAGA3, AGA7, and SUAGA7
- Added Tag Path validation to configuration
- No fixes
- No new features
- No fixes
- No new features
- No fixes

Release notes for earlier module versions available here