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

Cirrus Link Modules for Ignition 8.x

Expand/Collapse All to allow Search

- 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](#)