Archive Ignition 8.x Compatible Release Notes

Cirrus Link Modules for Ignition 8.x

Expand/Collapse All to allow Search

- No new features

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions
- Fixed bug that resulted in MQTT Distributor starting when actually disabled after a 'system disabled' restore

- Added support for applying sending Ignition SecurityContext based writes via Sparkplug
- Added support for tag latching in custom namespaces
- Added support for 'string conversions' in Sparkplug and custom namespaces including regex support

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions
- Fixed NPE in Engine handling of STATE message arrival
- Fixed tag configuration handling of properties in some edge cases
- Modified MQTT Engine to not update tags that are disabled at the Edge
- Fixed order of operations on start/stop when reconfiguring MQTT Engine
- Fixed UDT paramterized property handling in Engine

- Added support for applying Ignition SecurityContext based writes via Sparkplug
- Added BIRTH caching to fix in-order history across Transmission/Ignition restarts
- Added support for Transmission file publishing filenames to have date/timestamps
- Added a retry configuration option for file publishing
- Modified Transmission to allow file publishing even if commands are blocked
- Fixed Example Transmitter creation on Ignition Edge systems
- Improved disk-backed history store migration code
- Added support to enable/disable Server Settings
- Improved file publishing throughput
- Improved shutdown speed when lots of Edge Nodes exist in Transmission

- Fixed aliasing migration code to prevent malformed configurations
- Added support to delete Disk-backed history store artifacts when an Edge Node or Device is deleted
- Fixed 'Refresh Required' tag indicator handling
- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions
- Added null protection to configs that may omit Compression Type and Data Format Type
- Fixed support for 'Single level Edge Node refreshing' when a Group ID is specified in the Transmitter config
- Improved thread handling in start/stop scenarios when auto-publishing
- Fixed handling of parameterized UDT definition members
- Fixed handling of auto-publishing when not connected to primary host
- Fixed to file deletion handling when auto-publishing files
- Improved synchronization to start/stopping of threads on reconfiguration operations
- Fixes to file publishing thread start/stop
- Modified Transmission to not remove the 'formatString' property from UDTs
- Fixed initialization of RAM based history stores for some startup scenarios
- Fixed a thread leak on 'single level Edge Node refresh'
- Removed unnecessary thread startup of Edge Nodes that could result in invalid bdSeq numbers being published in an NBIRTH
- Fixed file publish retry logic

- No new features

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing version

- No new features

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing version

- Added support for Content Encoding to be specified for IoT Hub connections
- Added support for Azure IoT Central

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions
- Fixed certificate upload handling and validation

- Moved private key storage to the Ignition internal DB to improve usability
- Added support for HTTP and HTTPS proxies

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions

- No new features

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions

- Added 'Core Diagnostic' informational tags to the managed tag provider

- Deprecated/removed 'Poll EOT' and 'Auto-discovery EOT' timeouts
- Improved connection status indication in the Ignition Gateway Web UI
- Fixed null Station ID handling in configuration

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions
- Improved CSV config file handling
- Fixed mandatory field handling to improve the configuration experience
- Fixed md5sum calculations for configuration file handling

- Improved connection status indication in the Ignition Gateway Web UI
- Improved handling of 'Point type 0' TLPs

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions
- Fixed md5sum calculations for configuration file handling

- No new features

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions

- Modifications to not use 'default' (zero for numerics, false for booleans, and empty strings) values before an initial scan is performed
- Logging improvements

- Fixed potential class loading issues when running multiple Cirrus Link modules of differing versions

- No new features

- No fixes

- General logging improvements
- Modified MQTT Engine to only subscribe on Elecsys 'State' topics if that namespace is enabled
- Added dialog documentation for the configuration Python scripting API
- Added support to allow spaces in Primary Host IDs
- Added support for non-RSA type (PKCS#8) private keys

- Fixed tag writes for Document type tags
- Fixed server name validation to only allow characters that are valid in Ignition tag names
- Various NPE fixes around invalid UDT configurations
- Fixed issue that could result in MQTT Engine publishing Rebirth requests to the wrong MQTT Server in redundant environments

- Removed the 'Node Control/Reboot' metric from the NBIRTH as it doesn't make sense for Transmission
- General logging improvements
- Added validation to 'Tag Provider' and 'Tag Path' configuration fields
- Added support for 'Edge Node level Refresh' to Transmission along with Edge Node granularity 'Refresh Required' tracking
- Fixed server name validation to only allow characters that are valid in Ignition tag names
- Added dialog documentation for the configriuation Python scripting API
- Added support for 'Refresh Required' updates on tag property changes
- Fixed handling of Datasets in history stores
- Added support to allow spaces in Primary Host IDs
- Added support for 'discovering edge nodes' with corresponding 'discover required' tag and auto-removal of Edge Nodes in Transmitter tag trees
- Added support for non-RSA type (PKCS#8) private keys

- Fixed detection of tag change events on Document type tags
- Fixed potential (rare) deadlock when connecting to MQTT Servers
- Fixed potential NPE around null bound property values in UDTs
- Fixed issue where UDT null bound property values could be omitted from a payload
- Fixed potential deadlock on shutdown
- Fixed handling in Document tag types in DATA messages to prevent conversion to String datatypes in MQTT Engine
- Fixed issue in alias handling that could result in duplicate aliases
- Fixed potential NPE when handling a UDT definition member tag property bound to a UDT parameter
- Fixed publishing of UDT definitions in the NBIRTH when converting UDTs
- Fixed potential NPE that could occur in Docker startup environments when creating the 'Example Transmitter'
- Fixed bdSeq number handling that could result in it being incorrect in NBIRTH messages after connection failures
- Fixed potential NPEs during Transmitter shutdown
- Various Disk Backed history store fixes around NPEs and potential DB lockout
- Various NPE fixes around invalid UDT configurations

- No new features

- No fixes

- Added validation to 'Tag Provider' and 'Tag Path' configuration fields
- Added dialog documentation for the configriuation Python scripting API
- Added support for pushing DDEATH messages on Device removals

- Fixed detection of tag change events on Document type tags
- Fixed potential NPE around null bound property values in UDTs
- Fixed issue where UDT null bound property values could be omitted from a payload
- Fixed potential deadlock on startup
- Fixed potential NPE when handling a UDT definition member tag property bound to a UDT parameter

- Added support for Azure Edge configuration via the Python scripting API
- Added support for 'auto-configuration' in Azure Edge Docker environments
- Added validation to 'Tag Provider' and 'Tag Path' configuration fields
- Added dialog documentation for the configuration Python scripting API
- Added support for configurable character encoding
- Added support for pushing DDEATH messages on Device removals
- Added support for non-RSA type (PKCS#8) private keys

- Fixed detection of tag change events on Document type tags
- Fixed potential NPE around null bound property values in UDTs
- Fixed issue where UDT null bound property values could be omitted from a payload
- Fixed potential deadlock on startup
- Fixed potential NPE when handling a UDT definition member tag property bound to a UDT parameter

- Added validation to 'Tag Provider' and 'Tag Path' configuration fields
- Made session token expirations configurable
- Added dialog documentation for the configuration Python scripting API
- Added support for pushing DDEATH messages on Device removals
- Added support for non-RSA type (PKCS#8) private keys

- Fixed detection of tag change events on Document type tags
- Fixed potential NPE around null bound property values in UDTs
- Fixed issue where UDT null bound property values could be omitted from a payload
- Fixed potential deadlock on startup
- Fixed potential NPE when handling a UDT definition member tag property bound to a UDT parameter

- Added validation to 'Tag Provider' and 'Tag Path' configuration fields
- Added dialog documentation for the configuration Python scripting API
- Added support for pushing DDEATH messages on Device removals
- Added support for non-RSA type (PKCS#8) private keys

- Fixed detection of tag change events on Document type tags
- Fixed potential NPE around null bound property values in UDTs
- Fixed issue where UDT null bound property values could be omitted from a payload
- Fixed potential deadlock on startup
- Fixed potential NPE when handling a UDT definition member tag property bound to a UDT parameter

- Added a 'watchdog' to ensure continued AAR polling
- Added validators to global definition files
- Logging improvements around supervisory frame status
- Moved large configuration files from the internal Ignition DB to the filesystem which is supported in Ignition8 backup/restore operations
- Improved AAR write times
- Fixed issue that prevented Totalflow Application definitions from being uploaded via Windows in some scenarios

- Fixed cleanup of 'orphaned' global definition records
- Fixed import issue with array register templates
- Fixed issue with 'Station ID' character lengths
- Added support for comma characters in Totalflow Application's Name, Revision, and Parameter fields

- Added a 'watchdog' to ensure continued TLP polling
- Added validators to global definition files
- Added retry logic to login attempts when login fails
- Fixed issue where polling could stop in certain scenarios and not automatically restart
- Moved large configuration files from the internal Ignition DB to the filesystem which is supported in Ignition8 backup/restore operations

- Fixed cleanup of 'orphaned' global definition records

- No new features

- No fixes

- No new features

- No fixes

- Added support for Python script based configuration changes

- No fixes

- Improved handling of setting Edge Nodes and Devices offline
- Added support for Python script based configuration changes
- Improved general performance
- Added TLS ALPN support
- Improved logic around subscriptions that may fail due to ACL or permission issues
    - NOTE: This fix can result in MQTT Engine no longer being able to connect to the MQTT Server when attempting to subscribe on a topic without the permissions to do so. This is a configuration issue that needs to be fixed by either:
        - (1) Updating the MQTT Server ACL/permissions so Engine can subscribe on the required topics OR
        - (2) Disable MQTT Engine Namespaces and/or PrimaryHostId so that Engine is no longer required to subscribe on topics without the permission to do so.
- Improved processing speed of custom namespace JSON payloads

- Fixed issue that could result in Engine not sending commands when required including Rebirth requests
- Added protection for null payload timestamps
- Fixed issue that could result in deleted info tags associated with server settings
- Internal tag cache fixes and improvements
- Changed configureTag logic to only configure UDT instances when it is new

- Added support for Python script based configuration changes
- Added support for better QualityCode comparisons to support changes in Ignition 8.1.7
- Added TLS ALPN support
- Improved logic around subscriptions that may fail due to ACL or permission issues
    - NOTE: This fix can result in MQTT Transmission no longer being able to connect to the MQTT Server when attempting to subscribe on a topic without the permissions to do so. This is a configuration issue that needs to be fixed by either:
        - (1) Updating the MQTT Server ACL/permissions so Transmission can subscribe on the required topics OR
        - (2) Clear the MQTT Transmission PrimaryHostId so that Transmission is no longer required to subscribe on the STATE topic without the permission to do so.
- Improved history store startup logic to better handle mis-configurations

- Fixed 'Refresh Required' tag handling to show the proper state of the tag
- Internal cache fixes and improvements
- Fixed potential thread-leak
- Fixes and improvements to bound properties in UDTs
- Fixes to history store and forward when UDTs are being used

- No new features

- No fixes

- Added support for Python script based configuration changes

- Fixed potential thread-leak

- Added support for Python script based configuration changes
- Added support for Azure IoT Edge Hubs
- Added support for IoT Hub 'Application Properties' via Injector configuration

- Fixed potential thread-leak

- No new features

- Fixed potential thread-leak

- No new features

- Fixed potential thread-leak

- Improvements to connection and re-connection logic

- Fixed naming collision for 'periodic mappings' between ROC and ABB mappings
- Fixed bug regarding event polling in an edge case

- Improvements to connection and re-connection logic
- Added 'polling state' to the ROC devices

- Fixed naming collision for 'periodic mappings' between ROC and ABB mappings

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- Added support for MQTT Client IDs longer than 23 characters in length is user specified
- Improved error handling for unsupported characters in custom namespace tag creation
- Removed BB Wzzard namespace as the provided namespace as it can now be supported via a JSON custom namespace
- Fixed subscriptions to prevent potential duplicate messages based on MQTT server implementations
- Upgraded underlying MQTT library
- Added support for 'Document' tag datatype
- Added support for PermissionsConfig, TagEventScripts and ParameterValues
- Modified subscriptions on STATE messages to be on STATE/phid rather than STATE/#
- Improved processing speed of custom namespace JSON payloads
- Added support to handle new 'Optimized UDT' DATA messages

- Fixed datatype handling for sampleMode and historyTimeDeadbandUnits properties
- Fixed logic to only unsubscribe from STATE topics when appropriate

- Added code to send a rebirth request in the case DDATA messages arrive before DBIRTH
- Improved error handling in the case of missing bdSeq numbers
- Fixed topic token filtering bug betwen Sparkplug A and Sparkplug B namespaces
- Fixed issue where UDT definition member tag history configuration is overwritten when in a folder
- Fixed issue that could result in Engine not correcting stale 'STATE' messages when redundancy is used
- Fixed issue that could show RBE enabled tags stale when they were not actually stale
- Fixed bug that didn't show online/offiine node info dataset tags properly
- Fixed handling reporting of primary host state informational tags
- Fixed issue that prevented rebirth requests from being sent out when 'sequence reordering' is enabled

- Added support for MQTT Client IDs longer than 23 characters in length is user specified
- Added 'event tracing code' to track execution time for tag change events that occur
- Upgraded underlying MQTT library
- Added support for 'Document' tag datatype
- Added support for PermissionsConfig, TagEventScripts and ParameterValues
- Logging improvements around UDT handling
- Added bound value support for UDTs to evaluate bindings before publishing data
- Added a configurable reconnect delay to slow reconnect attempts on low bandwidth networks
- Added support for UDT parameter to UDT parameter bindings

- Fixed issue with potential failed file transfers when primary host is being used
- Fixed birth caching when aliasing is enabled
- Fixed issue where tags were not reported as stale when using 'Convert UDTs' and they were deleted
- Fixed Transmission to only publish RECORDs if the associated Sparkplug Device exists
- Fixed UDT definition handling to support definitions with the same name under different folder paths

- No new features

- No fixes

- Cleaned up AWS credential text in the configuration Web UI
- Added 'event tracing code' to track execution time for tag change events that occur
- Added support for 'Push to Any' option to 'Sets' as is supported for other Injector modules
- Added bound value support for UDTs to evaluate bindings before publishing data
- Increased the Kinesis default partition key count size as recommended by AWS
- Logging improvements around UDT handling
- Added support for 'periodic publishing' as an alternative to event/tag change driven publishing
- Added support for UDT parameter to UDT parameter bindings
- Added more randomness to partition keys per AWS recommendations to Kinesis partition keys
- Improved processing time on tag change events when events occur on UDT members
- Added 'Optimize UDT' option to Transmission to reduce payload sizes in DATA messages

- Fixed implementation to publish NDEATH message when Injector is disabled
- Fixed UDT definition handling to support definitions with the same name under different folder paths

- Added support for MQTT Client IDs longer than 23 characters in length is user specified
- Added 'event tracing code' to track execution time for tag change events that occur
- Upgraded underlying MQTT library
- Added content type and content encoding options to Azure Injector IoT Hubs to support message based routing
- Logging improvements around UDT handling
- Added a configuration option to support Azure Time Series Insights data/time format
- Added support for 'periodic publishing' as an alternative to event/tag change driven publishing
- Added bound value support for UDTs to evaluate bindings before publishing data
- Added support for UDT parameter to UDT parameter bindings
- Improved processing time on tag change events when events occur on UDT members
- Added 'Optimize UDT' option to Transmission to reduce payload sizes in DATA messages

- Fixed issue where disabled configurations of IoT Hub would still send BIRTH messages
- Fixed implementation to publish NDEATH message when Injector is disabled
- Fixed UDT definition handling to support definitions with the same name under different folder paths

- Added support for MQTT Client IDs longer than 23 characters in length is user specified
- Added 'event tracing code' to track execution time for tag change events that occur
- Upgraded underlying MQTT library
- Logging improvements around UDT handling
- Added support for 'periodic publishing' as an alternative to event/tag change driven publishing
- Added bound value support for UDTs to evaluate bindings before publishing data
- Added support for UDT parameter to UDT parameter bindings
- Improved processing time on tag change events when events occur on UDT members
- Added 'Optimize UDT' option to Transmission to reduce payload sizes in DATA messages

- Fixed implementation to publish NDEATH message when Injector is disabled
- Fixed UDT definition handling to support definitions with the same name under different folder paths

- Added support for MQTT Client IDs longer than 23 characters in length is user specified
- Added 'event tracing code' to track execution time for tag change events that occur
- Upgraded underlying MQTT library
- Logging improvements around UDT handling
- Added support for 'periodic publishing' as an alternative to event/tag change driven publishing
- Added bound value support for UDTs to evaluate bindings before publishing data

- Added support for UDT parameter to UDT parameter bindings
- Improved processing time on tag change events when events occur on UDT members
- Added 'Optimize UDT' option to Transmission to reduce payload sizes in DATA messages

- Fixed implementation to publish NDEATH message when Injector is disabled
- Fixed UDT definition handling to support definitions with the same name under different folder paths

- Initial GA release of the module for Ignition8

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- No new features

- No fixes

- Initial beta1 release of the SiteWise Engine Module for Ignition8

- No fixes

- Modified info tags to properly re-init them if they have been deleted

- Fixed handling of default configurations on initial module installation
- Fix to ensure client count displays zero when MQTT Distributor is shutdown
- Fixed to dynamic handling in configuration menus

- General logging cleanup
- Modified info tags to properly re-init them if they have been deleted
- Removed support for consumption of alarm properties
- Modified auto-RPC reconnect settings to denote whether or not to connect at all
- Added java.sql.Timestamp support to Ignition DataSet
- Added support for filtering by both Sparkplug group and edge node IDs
- Added support for most Ignition array datatypes in Ignition8
- Added primary host enable/disable control tag as well as a tag to specify the primary host ID
- Moved 'File handling control' of MQTT Engine to the Sparkplug B Namespace settings
- Expanded file handling support

- Fixed potential sequence reordering NPE
- Fixed history property coersion by using proper datatypes
- Fixed String vs Integer datatype handling for sampleMode and historyTimeDeadbandUnits
- Fixed potential Exceptions during server connection establishment
- Fixed handling of default configurations on initial module installation
- Fixed Enable Latency Check Tag to only sub/unsub when state changes
- Fixed [MQTTEngine]_types_/_types_/... bug in UDT definition creation
- Fixed to dynamic handling in configuration menus

- General logging cleanup
- Modified info tags to properly re-init them if they have been deleted
- Added primary host and BD sequence number validation on unexpected LWTs
- Improved disk-backed store and forward Exception handling
- Removed support for sending of alarm properties
- Modified auto-RPC reconnect settings to denote whether or not to connect at all
- Added java.sql.Timestamp support to Ignition DataSet
- Added support for 'ParameterValue' objects introduced in Ignition 8.0.15
- Added code to bind bound parameters of UDTs to tag properties if they are simple bindings
- Added 'cached birth' option to Transmission to improve Rebirth request performance
- Added partial support for overridden property values bound to UDT parameters
- Added support for most Ignition array datatypes in Ignition8
- Added the ability to transmit binary files from MQTT Transmission
- Made the Tag Pacing Period a mandatory field
- Added enforcement to prevent 'aliasing' and publishing of UDTs to be enabled at the same time
- Added code to specify a default 'Set' when appropriate during configuration
- Added code to ignore UDT removal events when convertUdts is true

- Added code to ensure Transmission connection verification accounts for random startup delay and debounce time
- Fixed UDT quality handling issues
- Fixed potential Exceptions during server connection establishment
- Fixed in-order history flushing bug when many devices are are part of an Edge Node that prevented buffer from ever emptying
- Fixed configureTag call in Transmission Record creation to properly configure the 'Publish Tag'
- Fixed handling of default configurations on initial module installation
- Fixed missing UDT member tag changes when TagPath is empty
- Added null check for null DataSet values on metrics
- Fixed UDT handling to extract the Tag provider name from when not found in the tag configuration to improve UDT handling

- Fixed thread leak in record publishes
- Cleaned up shutdown code to prevent RejectedExecutionExceptions
- Fixed UDT member tag property to UDT parameter bindings for 'convert to normal tags' case
- Fixed null value datatype coercion to prevent NPEs and tag reading
- Fixed disk-backed store and forwarding when aliasing is enabled
- Improved Exception handling during startup and iterating the tag tree
- Fixed to dynamic handling in configuration menus
- Fixed handling of parameterized datatypes
- Fix for ClassCastException on getting compression type
- Added Exception handling around adding parameters to UDTs to prevent UDT parsing failures that prevent them from being published

- Modified info tags to properly re-init them if they have been deleted

- Fixed handling of default configurations on initial module installation
- Fixed to dynamic handling in configuration menus

- Modified info tags to properly re-init them if they have been deleted
- Added java.sql.Timestamp support to Ignition DataSet
- Added support for 'ParameterValue' objects introduced in Ignition 8.0.15
- Added code to bind bound parameters of UDTs to tag properties if they are simple bindings
- Added partial support for overridden property values bound to UDT parameters
- Added support for most Ignition array datatypes in Ignition8
- Made the Tag Pacing Period a mandatory field
- Added code to specify a default 'Set' when appropriate during configuration

- Fixed UDT quality handling issues
- Fixed handling of default configurations on initial module installation
- Fixed missing UDT member tag changes when TagPath is empty
- Added null check for null DataSet values on metrics
- Fixed UDT handling to extract the Tag provider name from when not found in the tag configuration to improve UDT handling
- Cleaned up shutdown code to prevent RejectedExecutionExceptions
- Fixed UDT member tag property to UDT parameter bindings for 'convert to normal tags' case
- Improved Exception handling during startup and iterating the tag tree
- Fixed to dynamic handling in configuration menus
- Fixed handling of parameterized datatypes
- Fixed potential ClassCastExceptions
- Added Exception handling around adding parameters to UDTs to prevent UDT parsing failures that prevent them from being pushed

- Modified info tags to properly re-init them if they have been deleted
- Added configurable MQTT keep-alive setting
- Added java.sql.Timestamp support to Ignition DataSet
- Added support for 'ParameterValue' objects introduced in Ignition 8.0.15
- Added code to bind bound parameters of UDTs to tag properties if they are simple bindings
- Added support for client based auth in Azure IoT Hubs
- Added partial support for overridden property values bound to UDT parameters
- Added support for most Ignition array datatypes in Ignition8
- Made the Tag Pacing Period a mandatory field
- Added code to specify a default 'Set' when appropriate during configuration

- Fixed UDT quality handling issues
- Fixed potential Exceptions during MQTT server connection establishment
- Fixed handling of default configurations on initial module installation
- Fixed missing UDT member tag changes when TagPath is empty
- Added null check for null DataSet values on metrics
- Fixed UDT handling to extract the Tag provider name from when not found in the tag configuration to improve UDT handling
- Cleaned up shutdown code to prevent RejectedExecutionExceptions
- Fixed UDT member tag property to UDT parameter bindings for 'convert to normal tags' case
- Improved Exception handling during startup and iterating the tag tree
- Fixed to dynamic handling in configuration menus
- Fixed handling of parameterized datatypes
- Fixed potential ClassCastExceptions
- Added Exception handling around adding parameters to UDTs to prevent UDT parsing failures that prevent them from being pushed

- Modified info tags to properly re-init them if they have been deleted
- Added configurable MQTT keep-alive setting
- Added support for 'ParameterValue' objects introduced in Ignition 8.0.15
- Added code to bind bound parameters of UDTs to tag properties if they are simple bindings
- Fixed UDT member tag property to UDT parameter bindings for 'convert to normal tags' case
- Added partial support for overridden property values bound to UDT parameters
- Added support for most Ignition array datatypes in Ignition8
- Made the Tag Pacing Period a mandatory field
- Added code to specify a default 'Set' when appropriate during configuration

- Fixed UDT quality handling issues
- Fixed potential Exceptions during MQTT server connection establishment
- Fixed handling of default configurations on initial module installation
- Fixed missing UDT member tag changes when TagPath is empty
- Added java.sql.Timestamp support to Ignition DataSet
- Added null check for null DataSet values on metrics
- Fixed UDT handling to extract the Tag provider name from when not found in the tag configuration to improve UDT handling

- Cleaned up shutdown code to prevent RejectedExecutionExceptions
- Improved Exception handling during startup and iterating the tag tree
- Fixed to dynamic handling in configuration menus
- Fixed handling of parameterized datatypes
- Fixed potential ClassCastExceptions
- Added Exception handling around adding parameters to UDTs to prevent UDT parsing failures that prevent them from being pushed

- Modified info tags to properly re-init them if they have been deleted
- Added configurable MQTT keep-alive setting
- Added support for 'ParameterValue' objects introduced in Ignition 8.0.15
- Added code to bind bound parameters of UDTs to tag properties if they are simple bindings
- Fixed UDT member tag property to UDT parameter bindings for 'convert to normal tags' case
- Added partial support for overridden property values bound to UDT parameters
- Added support for most Ignition array datatypes in Ignition8
- Made the Tag Pacing Period a mandatory field
- Added code to specify a default 'Set' when appropriate during configuration

- Fixed UDT quality handling issues
- Fixed potential Exceptions during MQTT server connection establishment
- Fixed handling of default configurations on initial module installation
- Fixed missing UDT member tag changes when TagPath is empty
- Added java.sql.Timestamp support to Ignition DataSet
- Added null check for null DataSet values on metrics
- Fixed UDT handling to extract the Tag provider name from when not found in the tag configuration to improve UDT handling
- Cleaned up shutdown code to prevent RejectedExecutionExceptions
- Improved Exception handling during startup and iterating the tag tree
- Fixed to dynamic handling in configuration menus
- Fixed handling of parameterized datatypes
- Fixed potential ClassCastExceptions
- Added Exception handling around adding parameters to UDTs to prevent UDT parsing failures that prevent them from being pushed

- No new features

- No fixes

- Initial GA release

- No fixes

- Initial GA release

- No fixes

- If a controller tag is scanned as invalid, all of it’s expanded (un-optimized) root tags will be marked as stale. For example, if a table is invalid, all of it’s table elements will stale or an integer scanned becomes invalid, all of its boolean, bit-picked, and integer values will stale.
- Reconnect algorithm has been streamlined to minimize lag

- Tags are refreshed on each occurrence the scanner connects to the controller

- Initial release of module for Ignition 8

- No fixes

- Initial release of module for Ignition 8

- No fixes

- Initial release of module for Ignition 8

- No fixes

- Added support for Ignition Maker Edition
- Added support for breaking 'SecurityContext' API change in Ignition 8.0.13
- Added backward compatibility support so v4.0.5 will run on Ignition v8.0.12 and before

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration

- Added support for Ignition Maker Edition
- Added support for custom namespace charset selection for binary encoding support - set default to UTF-8
- Added support for breaking 'SecurityContext' API change in Ignition 8.0.13
- Added backward compatibility support so v4.0.5 will run on Ignition v8.0.12 and before
- Modified certificate display in Server configuration to not show any certs since we can't show all of them
- Improved handling for missing bdSeq numbers in faulty Sparkplug edge client's LWTs
- Added alarmEvalEnabled to default property filter list
- Added support for modified 'configureTag' behavior in Ignition 8.0.14

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration
- Fixed access to property matching for proper property filtering

- Added support for Ignition Maker Edition

- Added support for breaking 'SecurityContext' API change in Ignition 8.0.13
- Logging improvements
- Added backward compatibility support so v4.0.5 will run on Ignition v8.0.12 and before
- Added alarmEvalEnabled to default property filter list
- Added support for modified 'configureTag' behavior in Ignition 8.0.14
- Added code to ignore _invalid_ tag paths when traversing the tag tree

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration
- Fixed Transmission version reporting in the Transmission Info tag
- Fixed NPE when getting the module version
- Fixed 'reset metrics' info tag to always be writable

- No new features

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration

- Added support for breaking 'SecurityContext' API change in Ignition 8.0.13
- Added backward compatibility support so v4.0.5 will run on Ignition v8.0.12 and before
- Added code to ignore _invalid_ tag paths when traversing the tag tree

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration
- Fixed issue with null-check on S&F type
- Fixed 'Edge Nodes' folder creation to occur the the correct tag provider

- Added support for breaking 'SecurityContext' API change in Ignition 8.0.13
- Added backward compatibility support so v4.0.5 will run on Ignition v8.0.12 and before
- Added code to ignore _invalid_ tag paths when traversing the tag tree

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration
- Fixed issue with null-check on S&F type
- Fixed issue with 'Push Policy' not being persisted properly
- Fixed 'Edge Nodes' folder creation to occur the the correct tag provider

- Added support for breaking 'SecurityContext' API change in Ignition 8.0.13
- Added backward compatibility support so v4.0.5 will run on Ignition v8.0.12 and before
- Added code to ignore _invalid_ tag paths when traversing the tag tree

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration
- Fixed issue with null-check on S&F type
- Fixed issue with 'Push Policy' not being persisted properly
- Fixed 'Edge Nodes' folder creation to occur the the correct tag provider

- Added support for breaking 'SecurityContext' API change in Ignition 8.0.13
- Added backward compatibility support so v4.0.5 will run on Ignition v8.0.12 and before
- Added code to ignore _invalid_ tag paths when traversing the tag tree

- Fixed enum usage to prevent class loading exceptions when browsing/modifying gateway web UI configuration
- Fixed issue with null-check on S&F type
- Fixed issue with 'Push Policy' not being persisted properly
- Fixed 'Edge Nodes' folder creation to occur the the correct tag provider

- No new features

- No fixes

These modules are not compatible Ignition 8.0.13 or later. If using 8.0.13 or 8.0.14 use v4.0.5 Cirrus Link modules

- Updated code signing cerificate

- Fixed documentation links in tooltips of the module config pages

- Modified TLS setup to piggyback off of native Ignition SSL mechanism

- No fixes

- Add custom property for 'sequence number reordering'
- Modified to create default reasonable history config on all dynamic MQTT Engine tags
- Added a validator to primary host ID creation to prevent runtime errors
- Added support for non-clean session and QoS1 support in custom namespaces
- Added additional tooltip info for Custom Namespace writable tags
- Improved certificate file management in the config web configuration UI
- Fixed 'Critical App State' info tag to prevent erroneous data
- Added a cache layer for improved custom namespace performance

- Fixed retention of UDT properties across module restarts
- Fixed client reinitialization after custom namespace changes
- Fixed Edge Node Tag staling from NDATA message
- Added staling of custom namespace tags on namespace deletion
- Fixed 'empty' property filter setting to allow all

- Added a validator to primary host ID creation to prevent runtime errors
- Added parameter based UDT DataType handling when convertUDTs=true
- Added tag pacing period validator to ensure it is greater than zero
- Improved certificate file management in the config web configuration UI
- Added Transmission Info Tags for tracking messsages and bytes

- Fixed dynamic update of client ID under MQTT Transmission Info tags
- Added support to allow RPC clients to not automatically reconnect
- Improved DB reconnect logic for the Disk-backed store and forward mechanism
- Fixed Timer thread leak on pending publish creation
- Fixed to caching of Transmission Info metrics to prevent erroneous data
- Fixed handling of non-memory tags (i.e. derived, OPC, etc) in Sparkplug Record publishes
- Fixed potential ClassCastException around BoundPropertySet Objects
- Fixed Transmission TLS connect on startup
- Fixed NPE from caching BIRTHs when TransmissionClient is shutdown
- Removed properties and parameters from being improperly published in DATA messages
- Fixed missing Sparkplug IDs on Transmitters page
- Added code to ignore _types_ folder when creating Edge Nodes
- Fixed handling of EdgeNode tag removal events
- Added new custom property 'ConfiguredTagPath' to tag properties
- Fixed RPC clients to update on server setting change
- Fixed 'empty' property filter setting to allow all
- Fixed fetching of the tag provider name by ignoring case
- Fixed device detection when auto-discovery is enabled
- Fixed NPE on aliased metrics in NCMD messages
- Fix for Tag add/remove root path comparisons
- Fixed folder detection on tag structure change events
- Fixed check history stores to update Transmitters on configuration changes

- No new features

- No fixes

- Added tag pacing period validator to ensure it is greater than zero
- Disconnect logic cleanup
- General performance enhancements

- Fixed Timer thread leak on pending publish creation
- Fixed potential ClassCastException around BoundPropertySet Objects
- Added null checks to Injector start/stop to prevent NPEs
- Removed properties and parameters from being improperly published in DATA messages
- Added code to ignore _types_ folder when creating Edge Nodes
- Fixed handling of EdgeNode tag removal events
- Fixed device detection when auto-discovery is enabled
- Fix for Tag add/remove root path comparisons
- Fixed folder detection on tag structure change events

- Tooltip/help improvements
- Added tag pacing period validator to ensure it is greater than zero
- Removed unused certificate file upload field
- Added support for a connection pooling to allow for increased throughput
- Disconnect logic cleanup
- General performance enhancements

- Fixed Timer thread leak on pending publish creation
- Fixed potential ClassCastException around BoundPropertySet Objects
- Added null checks to Injector start/stop to prevent NPEs
- Removed properties and parameters from being improperly published in DATA messages
- Added code to ignore _types_ folder when creating Edge Nodes
- Fixed handling of EdgeNode tag removal events
- Fixed device detection when auto-discovery is enabled
- Fix for Tag add/remove root path comparisons
- Fixed folder detection on tag structure change events
- Added reconnect logic to MQTT Injectors when message pushes fail

- Added tag pacing period validator to ensure it is greater than zero
- Added support for a connection pooling to allow for increased throughput
- General performance enhancements
- Disconnect logic cleanup

- Fixed Timer thread leak on pending publish creation
- Fixed potential ClassCastException around BoundPropertySet Objects
- Added null checks to Injector start/stop to prevent NPEs
- Removed properties and parameters from being improperly published in DATA messages
- Added code to ignore _types_ folder when creating Edge Nodes
- Fixed handling of EdgeNode tag removal events
- Fixed device detection when auto-discovery is enabled
- Fix for Tag add/remove root path comparisons
- Fixed folder detection on tag structure change events
- Added reconnect logic to MQTT Injectors when message pushes fail

- Added tag pacing period validator to ensure it is greater than zero
- Added support for a connection pooling to allow for increased throughput
- General performance enhancements
- Disconnect logic cleanup

- Fixed Timer thread leak on pending publish creation
- Fixed potential ClassCastException around BoundPropertySet Objects
- Added null checks to Injector start/stop to prevent NPEs
- Removed properties and parameters from being improperly published in DATA messages
- Added code to ignore _types_ folder when creating Edge Nodes
- Fixed handling of EdgeNode tag removal events
- Fixed device detection when auto-discovery is enabled
- Fix for Tag add/remove root path comparisons
- Fixed folder detection on tag structure change events
- Added reconnect logic to MQTT Injectors when message pushes fail

- No new features

- No fixes

- Improved logging
- Improved BD sequence number handling
- Improved event handling
- Improved CirrusClient disconnection and LWT handling
- Updated auto-client ids to be 23 characters or less (required by some MQTT server implementations)

- No fixes

- No new features

- Fixed potential deadlock on shutdown
- Added Distributor Info tags showing number of connected clients

- Added support to records to pull data from ManagedTagProviders
- Improvements in Edge/Device cache
- Added support for 'writable tags' in Custom Namespaces
- Improvements in Metric caching

- Added staling of MQTT Engine Tags on startup
- Fixed various staling issues affecting MQTT Engine Tags
- Fixed issues with MQTT Engine restarts and deleted tags reappearing
- Fixed MQTT Engine client subscription handling to only subscribe to topics STATE/# and SM/<client_id>/# when necessary
- Fixed potential concurrency issues in MQTT Engine on reconnect
- Fixed the enabling/disabling of Namespaces on a configuration update

- Performance improvements in disk-backed history store and forward
- Added optional 'random' delay setting to Transmitter configuration

- Various fixes in disk-backed history store and forward
- Fixed potential thread leaks
- Fixed issue with BIRTH incorrectly containing cached/old Tag values

- Initial release

- No fixes

- Added settings validators to ensure correct Injector configuration

- No fixes

- Added settings validators to ensure correct Injector configuration

- Fixed shutdown issue to prevent duplicate MQTT client connections

- Added settings validators to ensure correct Injector configuration

- No fixes

- Added settings validators to ensure correct Injector configuration

- No fixes

- No new features

- No fixes

- Security improvements
- Added validation to ensure passwords can not be null, empty, or only whitespace
- Added ability to delete certificates from configurations
- Updated all existing password/secret configuration fields to use the "enter twice" validation

- Various minor bug fixes

- Add support for spaces in ACL topics

- No fixes

- Added support for updating/changing incoming tag datatypes
- Added 'Hostname Verification' setting to optionally enable TLS hostname verification
- Added support for configuring which Tag properties get filtered/ignored on incoming publishes
- Improved tracking of BIRTH/DEATH counts

- Fixed Custom Namespace support for periods (".") in MQTT topics, now interpreted as folder separator in the resulting tag path
- Fixed "Online Nodes" dataset initialization error
- Fixes for handling legacy quality types

- Replaced "Default Transmitter" with new "Example Transmitter" and associated example Tags.
- Made the publishing of UDT definitions in BIRTH messages configurable
- Added 'Last Refresh' Tag to denote time of last Refresh button click
- Added 'Refresh Required' Tag to denote if an MQTT Transmission refresh is required/needed
- Added protection against configuring the same History Store setting on multiple Transmitters
- Added 'Hostname Verification' setting to optionally enable TLS hostname verification
- Added support for configuring which Tag properties get published

- Fixed publishing of derived tags, OPC tags, and other indirect types

- Added support for configuring which Tag properties get published

- No fixes

- Added support for configuring which Tag properties get published

- No fixes

- Added support for configuring which Tag properties get published

- No fixes

- Added support for configuring which Tag properties get published

- No fixes

- Added settings validators to ensure correct Injector configuration

- No fixes

- Added scanner refresh control Tag

- No fixes

- No new features

- No fixes

- Added support for Ignition7/Transmission to publish to Ignition8/Engine with quality code translations

- Fixed online status info tags to denote the proper values
- Fixed multiple tag write issues
- Various tag path handling fixes
- Fixed issues with property merging in MQTT Engine
- Various UDT fixes
- Fixed typing of JSON tags in custom namespaces

- Improved Transmitter field validation in the Web UI configuration
- Added support for multiple tag paths in a single Transmitter configuration
- Added support for additional topic tokens to be published via special char sequence in a Sparkplug Device folder
- Improvements to handling of null values
- Added batching and caching components to improve performance

- Fixed multiple tag write issues
- Fixed support for auto-detected Edge Nodes in tag trees
- Fixed issue that could result in Transmitter Threads running after Transmission shutdown
- Various tag path handling fixes
- Various UDT fixes
- Fixed connected Server URL info tag handling
- Added sorting of UDT definitions to ensure proper order for creation of UDTs
- Fixed alias support
- Improvements to Disk-backed history stores

- Improvements to handling of null values
- Added Agent record validation in Web UI configuration

- Fixed support for auto-detected Edge Nodes in tag trees
- Various tag path handling fixes
- Various UDT fixes

- Improvements to handling of null values
- Added Agent record validation in Web UI configuration

- Fixed support for auto-detected Edge Nodes in tag trees
- Various tag path handling fixes
- Various UDT fixes
- Fixed publishing of NBIRTH messages for Event Hubs

- Improvements to handling of null values
- Added Agent record validation in Web UI configuration

- Fixed support for auto-detected Edge Nodes in tag trees
- Various tag path handling fixes
- Various UDT fixes

- Improvements to handling of null values
- Added Agent record validation in Web UI configuration

- Fixed support for auto-detected Edge Nodes in tag trees
- Various tag path handling fixes
- Various UDT fixes

- No new features

- No fixes

- Inital port to Ignition8

- No fixes

- Initial port to Ignition8
- Known issue that throughput is not as good as it was in Ignition 7 compatible modules

- No fixes

- Initial port to Ignition8

- No fixes

- Initial port to Ignition8

- No fixes

- Initial port to Ignition8

- No fixes

- Initial port to Ignition8

- No fixes

- Initial port to Ignition8
- Improved tag writing throughput working with massive tag lists and during episodes of congested writes

- An initially scanned floating point value of Not-A-Number (NaN) is better handled
- Scanner waits for acceptance of tags before publishing data
- Duplicate tag definitions are prevented
- In extreme cases, 32- and 64-bit integer values may have erratically updated