vNode v.1.22.8

Release Notes

Fixes, changes and features in vNode.

Released November 17, 2026

Platform (v.1.22.8)

Change	Description
Changed	CLI – The vNode executable now validates its command-line arguments and, when a command or argument is invalid, shows the error together with the usage and examples instead of failing with an exception.
Changed	Runtime – Updated the runtime from NodeJS 22.23.4 to NodeJS 26.13.1, which includes the V8 14.6 JavaScript engine.
Changed	Scripts – Node.js APIs removed in the new runtime are no longer available to Scripting module scripts and the npm libraries they use: the util.is* type checks (such as util.isString() or util.isNumber()), util._extend(), util.log(), process.assert(), SlowBuffer, fs.F_OK, fs.R_OK, fs.W_OK and fs.X_OK (use fs.constants), dirent.path (use dirent.parentPath) and the _stream_* modules. url.parse() and timers with a negative delay now log a deprecation warning. Action required: review the scripts and npm libraries of the Scripting module that use these APIs. ⓘ Scripting module scripts and npm libraries that use the removed Node.js APIs fail after the upgrade. Replace them with their current equivalents (for example typeof checks instead of util.is*, fs.constants instead of fs.F_OK, and the URL class instead of url.parse()).
Changed	Service – On Linux, when the system kills a single module for running out of memory (OOM), the vNode service now keeps running instead of stopping and restarting all modules. Action required: reinstall the vNode service on existing installations to apply this change.
Changed	Performance – Improved logging performance with faster timestamp formatting and batched writes to the log files.
Features	Scripts – Custom scripts can use the JavaScript features of the new runtime, such as the Temporal API, Float16Array, RegExp.escape(), Error.isError(), explicit resource management (using), Map.prototype.getOrInsert(), Iterator.concat(), base64 and hex conversion in Uint8Array, and the global URLPattern.
Features	Certificates – Modules can now store certificate files uploaded from their certificate views in Web UI, for example the OPC UA Server certificates.
Removed	Platforms – Removed support for Windows Server 2012 and 2012 R2, which the NodeJS 26 runtime no longer supports. Action required: upgrade these servers to Windows Server 2016 or later before upgrading vNode. ⓘ vNode 1.22.8 does not support Windows Server 2012 or 2012 R2. Upgrade to Windows Server 2016 or later before installing this version.

Platform (v.1.22.8)

Change	Description
Removed	Platforms – Removed support for Linux distributions with glibc earlier than 2.31: Debian 10, RHEL, AlmaLinux, Rocky Linux and Oracle Linux 8, Amazon Linux 2, and Raspberry Pi OS based on Debian 10 (Buster). Action required: upgrade these systems to a supported version, such as Debian 11, RHEL 9 or Amazon Linux 2023, before upgrading vNode. vNode 1.22.8 requires glibc 2.31 or later and does not start on Debian 10, RHEL, AlmaLinux, Rocky Linux or Oracle Linux 8, Amazon Linux 2 or Raspberry Pi OS Buster, where it reports "version 'GLIBC_2.31' not found". Check the glibc version with ldd --version and upgrade the system before installing this version.
Security	Runtime – Updated NodeJS to version 26.13.1 to incorporate the latest security patches.
Security	TLS – Secure connections now use the OpenSSL 3.5 default security level 2: RSA, DSA and DH keys shorter than 2048 bits, ECC keys shorter than 224 bits and RC4 cipher suites are rejected. Action required: replace certificates and keys below these sizes, both in vNode (Web UI HTTPS and module certificates) and in the devices, brokers, databases and servers it connects to. TLS connections and certificates that use RSA, DSA or DH keys shorter than 2048 bits, ECC keys shorter than 224 bits or RC4 cipher suites are rejected. Replace them with stronger keys before upgrading.
Security	TLS – Connections to a host written as a non-canonical IP address, such as 127.1, 0x7f.0.0.1, 2130706433 or 192.168.001.010, are now rejected with ERR_TLS_CERT_ALTNAME_INVALID instead of being matched against the server certificate (CVE-2026-106069, CVE-2026-106072, CVE-2026-105483). Action required: use canonical dotted-decimal IP addresses (for example 192.168.1.10) or host names in the connection settings of TLS connections. TLS connections whose host is configured as a non-canonical IP address (shortened, with leading zeros, hexadecimal or as a single integer) fail with ERR_TLS_CERT_ALTNAME_INVALID. Write the address in canonical dotted-decimal form or use the host name.

BACnet Client (v.1.3.0)

Change	Description
Changed	Browse – Properties that the BACnet standard defines as read-only are now shown and created as read-only tags. Priority_Array is always read-only.
Changed	Values – TIME and DATETIME values now include hundredths of a second (.CC), and DATE patterns are published as tokens such as '*-*-*', '2026-05-*' or '2026-odd-last'. Action required: update the systems that read these tag values if they parse the previous formats; the previous formats are still accepted on write.
Changed	Tags – Tag quality is now published only when it changes, and quality-only updates no longer include a value.

BACnet Client (v.1.3.0)

Change	Description
Features	COV – Added per-device Change-of-Value (COV) subscriptions. Subscriptions are cancelled when the module stops and renewed when a device answers again, and a COV lifetime below 30 seconds is raised to 30 with a warning.
Features	Data Structures – Added support for complex BACnet properties as structure or JSON string tags, including drag & drop of whole objects and structured properties into a nested tag tree and writes to individual fields.
Features	Data Types – Added support for the vNode number sub-types (int8 to float64) when creating and writing tags. REAL values are created as float32.
Features	Object Types – Added the object types and property identifiers of the ASHRAE 135-2020 addenda.
Fix	Values – Fixed an issue where DATE and TIME patterns (wildcards, odd or even month, last day, day of week) were published as 1900-01-01 or as a concrete time, and were written back with a different meaning.
Fix	Values – Fixed an issue where writes silently altered values instead of rejecting or encoding them correctly. Affected writes included object identifiers of another object type, enumerated or bit string properties without a known map, DATETIME values without a time, and invalid inputs such as '2024-02-30' or 1.9 for an unsigned value.
Fix	Configuration – Fixed an issue where an object instance or Device ID outside 0–4194303 silently addressed a different object. These fields, the property identifier, the routed network and the MAC address are now validated, and invalid tags or devices get BAD_CONFIG_ERROR quality.
Fix	Devices – Fixed an issue where disabled channels or devices still polled and wrote, a manual address was replaced by an I-Am from another address, and isolated timeouts eventually demoted a healthy device. Removed objects also kept being polled, and points missing from a truncated ReadPropertyMultiple answer stayed with good quality.
Fix	Writes – Fixed an issue where queued writes could reach the device in reverse order, and where pending writes never completed when the channel stopped.
Fix	Communication – Fixed an issue where a device that stopped answering delayed reads and writes to the other devices of the channel for minutes. Channel time is now shared among devices, and demoted devices are no longer polled, subscribed or written.
Fix	Discovery – Fixed an issue where a device that changed its IP address was not found again until the module restarted, and where channels bound to a specific interface on Linux did not receive broadcast I-Am messages.
Fix	Browse – Fixed an issue where Browse Devices listed the same device several times, mixed up answers on large networks and ignored Device ID 0.
Fix	Browse – Fixed an issue where proprietary object types (128 and above) could not be used as tags.

Bootstrap (v.1.49.0)

Change	Description
Changed	Logging – The "Running expression" message of custom condition alarms is now logged at TRACE level instead of DEBUG, reducing log noise when debug logging is enabled.
Changed	Alarms – Improved the performance of alarm requests (get, count and shelved alarms) on large tag models. Requests no longer scan every tag, significantly reducing CPU load when clients such as Web Vision poll alarms frequently.
Changed	LDAP – The connection check when the configuration loads no longer binds to the server, and "Invalid connection" is now logged only for network, TLS or timeout failures. When a login fails for one of these reasons, the log now shows the cause; users still see "Invalid credentials".
Changed	LDAP – Changing the LDAP host, port or TLS settings now closes the previous connection.
Fix	Logs – Fixed an issue where opening the realtime logs of the same module more than once stopped log delivery to the earlier view, and closing one of them left the others without logs.
Fix	LDAP – Fixed an issue where LDAP logins and configuration reloads hung after the domain controller recovered from an outage, until Bootstrap was restarted.
Fix	LDAP – Fixed an issue where the group search failed for users whose distinguished name (DN) contains parentheses.
Fix	Modules – Fixed an issue where a start that arrived while a module was stopping, for example during a restart, a redundancy transition or a monitor restart, could leave two processes of the module running or leave the module stopped until it was restarted again.
Fix	Store & Forward – Fixed an issue where changing the configuration of a required module paused Store & Forward even while the module was connected, and deleting a required module could leave Store & Forward paused for good.
Fix	Redundancy – Fixed an issue where switching to standby and stopping the service no longer waited for the modules to disconnect.
Fix	Redundancy – Fixed an issue where file synchronization with the backup node stopped if a changed file was deleted right before being read, until the redundancy reconnected or vNode was restarted.
Fix	Alarms – Fixed an issue where alarms unshelved together in a single request were shelved again after a restart when one of them had no saved shelving.
Fix	Tags – Fixed an issue where a tag received through Links that was then defined locally with the same path, or taken over by another link, kept the value, quality, timestamp and alarm state of its previous owner. Its alarms could also remain described as "[Link 'null' is down]".

CAISO ADS Client (v.1.5.0)

Change	Description
Changed	Logging – Communication warnings now include the underlying cause reported by the network or the CAISO ADS server, such as connection refused, timeouts or SOAP fault details, making connection issues easier to diagnose.
Security	Authentication – The client PFX certificate is now stored encrypted in the configuration and its backups instead of in plain form. Existing configurations keep working; select the PFX file again to encrypt the stored certificate.
Fix	Client – Fixed an issue where a single client with an invalid configuration, such as a missing certificate, prevented all other clients in the module from starting. The invalid client is now skipped and the error is logged.
Fix	Certificates – Fixed an issue where the client certificate or CA certificate failed to load when the selected file name contained the text “base64,”.

Data Exporter (v.1.11.0)

Change	Description
Changed	Dates – ISO 8601 dates in CSV, XML and JSON files now include milliseconds and end with Z under UTC, or with the zone offset under Server and Custom time zones. Previously they had no milliseconds and ended with +00:00 under UTC. Action required: update the systems that read the generated files if they parse these dates with a fixed pattern, or select the Custom date format ‘YYYY-MM-DDTHH:mm:ssZ’ to keep the previous output.
Changed	Configuration – Date options were moved into a dedicated Date section. Existing configurations are migrated automatically, and date fields not stored by older versions get their defaults (ISO 8601, UTC).
Changed	Formatting – The count and good aggregations are now written as plain numbers under every Format (for example, 2 instead of 2.00), and good is written at full precision. Action required: update the systems that read the generated files if they expect a fixed number of decimal places for these values.
Changed	Logging – Configuration problems are now reported as a single “Skipping agent” warning per agent with the cause. SFTP connection failures are logged as warnings at every Retry interval, and data-quality warnings name each affected tag once per export instead of once per value.
Features	File Types – Added JSON as an export file type, generating one document per group and interval with typed values, dates following the Date configuration, Compact or Extended Raw structure, and minified or pretty output.
Features	File Types – Added Parquet as an export file type. Aggregated files have one numeric column per aggregation and Raw files keep each value in a typed column. Dates are typed timestamps, values keep full precision, and Snappy, gzip or no compression can be selected.
Security	SFTP – The SFTP private key is now stored encrypted in the configuration and its backups instead of in plain form. Existing configurations keep working; select the key file again to encrypt the stored key.

Data Exporter (v.1.11.0)

Change	Description
Fix	Agents – Fixed an issue where an agent with an invalid configuration stopped the whole module. The agent is now skipped with a warning, and the other agents keep running.
Fix	Configuration – Fixed an issue where the module crashed when started with no saved configuration. It now logs an error and stays idle until a configuration is saved.
Fix	SFTP – Fixed an issue where an unreadable SFTP private key, such as a certificate selected instead of the key, crashed the module. The uploader now logs a warning, keeps the files pending and retries at the configured Retry interval.
Fix	Tags – Fixed an issue where a tag model subscription error crashed the module. The error is now logged and the module stops cleanly.
Fix	Formatting – Text tag values under a numeric Format now print the number they contain, or NaN when there is none, instead of 0 or 0.00. Object tag values print NaN under numeric Formats and their JSON text under %s, instead of [object Object]. Action required: update the systems that read the generated files if they rely on the previous 0, 0.00 or [object Object] output.
Fix	CSV – Fixed an issue where aggregated cells containing the separator, quotes or line breaks broke the CSV row. Cells now follow standard CSV escaping. Action required: update the systems that read the generated CSV files if they do not support standard CSV quoting.
Fix	Raw Export – Fixed an issue where a Format that failed on a Raw value stopped the export. The value is now written as plain text and a warning is logged.

DLMS Client (v.1.6.0)

Change	Description
Changed	Load Profile – After a module restart, the first load profile read now requests the last entry again, since it is usually flagged as erroneous or missed.
Changed	Load Profile – Load profile tags added while the module is running are now read through the scheduled request, right after any read in progress, so all the profile tags receive the same entries.
Fix	Clock – Fixed an issue where dates sent by the meter with their own deviation lost their time zone, so the meter's local time was published almost as UTC (one hour off for CET meters).
Fix	Load Profile – Fixed an issue where the load profile request range of devices with only load profile tags was shifted by the meter's time zone, because the clock was not synchronized before the request.
Fix	Load Profile – Fixed an issue where load profiles were read every two cron intervals instead of every interval.
Fix	Load Profile – Fixed an issue where two load profiles of the same meter overwrote each other's next start date, so each read started from the point reached by the other profile.

DLMS Client (v.1.6.0)

Change	Description
Fix	Load Profile – Fixed an issue where the load profile read stopped after a failed read until the device restarted, kept running after the device was stopped or its profile tags were removed, and applied the result of a read cancelled while it was in progress.
Fix	Load Profile – Fixed an issue where a misaligned entry (for example an event capture at 10:07) shifted the request range so the aligned entries were discarded, and meters that treat the range limits as exclusive did not return the first and last entries. The range is now aligned to the capture period and widened by one period at each side.
Fix	Load Profile – Fixed an issue where timestamp tags with numeric string values were ignored and, with several timestamp tags, the last one added was used instead of the most recently updated one. A new timestamp tag now takes the value of the existing ones, and the next start date is resolved from the timestamp tags, then the last read, then the persisted value.
Fix	Instant Values – Fixed an issue where instant readings stopped after a device error until the association failed or the channel reconnected.

DNP Server (v.1.5.2)

Change	Description
Fix	Protocol – Fixed an issue where requests with an unsupported application function code were answered with a malformed frame that DNP3 masters discarded, so they never received the “function code not supported” indication. The outstation now sends a complete response with the IIN bits set, and requests that do not expect a reply are ignored.
Fix	Unsolicited Responses – Fixed an issue where a Disable Unsolicited request was handled as a Read, returning the buffered class 1, 2 and 3 events and emptying the event buffers. The outstation now replies with a null response.

Fleet Manager Client (v.1.3.0)

Change	Description
Changed	Logging – Tunnel openings and closings are now logged with the tunnel name, destination and closing reason. No part of the token is written to the log.
Changed	Logging – Refused connections, registrations and status updates now explain the likely cause, such as a disabled or unknown token or the node UID, instead of showing a bare error code. Repeated refusals are summarized instead of being logged every time.
Changed	Logging – An invalid tunnel or module configuration is now reported in a single line that names the failing field, and a node without user tunnels no longer logs a warning at startup.
Security	Protocol – Limited memory usage when decoding messages with unknown fields, preventing excessive memory consumption from malformed or unexpected messages.

Fleet Manager Client (v.1.3.0)

Change	Description
Security	Dependencies – Updated the ws and ip-address libraries to address known security advisories.
Fix	Connection – Fixed an issue where the node did not reliably recover its session and status after a link or server failure. It now reconnects with back-off, checks the session as soon as the link is back, even with a slow broker, and sends its status right after every handshake.
Fix	Remote Commands – Fixed issues where remote commands never timed out, ran twice when redelivered, hung after a link drop, or were dropped after an unanswered command with the same identifier.
Fix	Tunnels – Fixed an issue where tunnels were not always released when they closed, particularly after a link drop, and tunnel data could be reordered or processed more than once after a reconnection.
Fix	Tunnels – Fixed an issue where a malformed tunnel frame crashed the module.
Fix	Tunnels – Fixed an issue where failed or unknown tunnel requests were reported to Fleet Manager as successful. They are now reported as errors.
Fix	Tunnels – Fixed an issue where a tunnel that could not be offered prevented the node from registering in Fleet Manager.
Fix	Configuration – Fixed an issue where an invalid tunnel or Web UI setting prevented the rest of the configuration from loading.
Fix	Startup – Fixed an issue where the module exited when the platform API was not ready or the configuration was invalid, and where stopping the module during startup still connected the node or made the stop fail.

Historian (v.1.25.1)

Change	Description
Security	TLS – The client private key used for TLS connections to external MongoDB and TimescaleDB databases is now stored encrypted in the configuration and its backups instead of in plain form. Existing configurations keep working; select the key file again to encrypt the stored key.
Fix	Store & Forward – Fixed an issue where draining a large Store & Forward (S&F) backlog into MongoDB could make the module run out of memory. Samples are now inserted in limited batches.
Fix	MongoDB – Fixed an issue on Linux where the embedded MongoDB database kept restarting in a loop after the Historian stopped unexpectedly, for example when killed by the system for lack of memory. The leftover database process is now stopped before the database starts again.
Fix	MongoDB – Fixed an issue where stopping the module could leave the embedded MongoDB database running.

Historian (v.1.25.1)

Change	Description
Fix	MongoDB – Fixed an issue where corrupt samples stored in MongoDB were returned as valid values by raw historical queries. They are now skipped and logged as a warning, and the rest of the data is still returned.
Fix	Timescale – Fixed an issue where historization to TimescaleDB stopped without reporting errors when the database came back after several module restarts or tag changes while it was unreachable.
Fix	Store & Forward – Fixed an issue where an event cut short at the end of a disk chunk could be delivered truncated as a valid event. Action required: before downgrading Historian to an earlier version, wait until the Store & Forward buffer is empty; earlier versions discard the pending data written by this version.

ICCP Server (v.1.0.2)

Change	Description
Security	Protocol – Fixed several issues where malformed or unexpected requests from an ICCP client could crash the server, including malformed association requests, transfer sets with an invalid data set scope, name list requests for unknown domains and malformed variable list deletions. These requests are now rejected or answered with an error, and the server keeps running.
Security	Dependencies – Updated the Tokio and error-stack libraries to address security advisories RUSTSEC-2026-0384 and RUSTSEC-2026-0198, and replaced unmaintained dependencies.
Fix	Data Sets – Fixed an issue where reading a data set that contained Next_DSTransfer_Set crashed the server. As required by IEC 60870-6-503, Next_DSTransfer_Set is no longer accepted as a data set member, in either client-defined or configured data sets.
Fix	MMS – Fixed an issue where responses and information reports could exceed the MMS PDU size negotiated with the client. Oversized responses are now answered with a resource error, and transfer sets are checked against the negotiated size when they are started.
Fix	Authentication – Fixed an issue where password authentication rejected clients that sent the authentication requirement in a longer bit string.
Fix	Connection – Fixed an issue where the session accept message sent to clients that omit the Called Session Selector declared an incorrect length.

IEC 102 Client (v.1.9.2)

Change	Description
Fix	Load Curve – Fixed an issue where adding a new timestamp tag while the module was running set it to the current time and discarded a start date that the user had just written in another timestamp tag, so the requested re-read of past data was lost. New timestamp tags now take the value of the existing ones.

MCP Server (v.2.0.0)

Change	Description
Changed	Access – Each token now lists the built-in tools, module methods and user-defined tools, prompts and resources it exposes, replacing the None, Read and Read-Write category levels. Action required: existing Read and Read-Write categories are migrated when the configuration is loaded (save it to keep the result); the user-defined tools, prompts and resources each token may use must be selected again. Token permissions no longer use the None, Read and Read-Write category levels. Read and Read-Write categories are migrated when the configuration is loaded; save the configuration to keep the result, and select again the user-defined tools, prompts and resources each token may use, or its clients lose access to them.
Changed	Tools – Tool responses now use minified JSON and compact tables to reduce the tokens read by the LLM. tag_history adds a 'decimals' argument and its metadata defaults to 'none', and backup_export saves the ZIP file in the exports folder and returns its path. Action required: update the systems that parse tool responses if they read the previous formats. Tool responses now use minified JSON and compact tables, tag_history metadata defaults to 'none', and backup_export returns the path of the ZIP file saved in the exports folder. Update the systems that parse tool responses if they read the previous formats.
Changed	Tools – Redesigned tool schemas and descriptions for small local models: lists are arrays, dates use ISO 8601, and allowed values and limits are published as schema enums and bounds. Common client input mistakes are accepted. Action required: update clients that send comma-separated values for list arguments to send arrays instead. List arguments now expect arrays. Update clients that send comma-separated values for list arguments to send arrays instead.
Changed	Tools – Tool calls with undeclared arguments are now rejected with an error naming the argument instead of ignoring it. Tool calls that include arguments not declared in the tool schema now fail with an error instead of ignoring them. Remove the extra arguments from clients and prompts.

MCP Server (v.2.0.0)

Change	Description
Changed	Alarms – alarm_get, alarm_count and alarm_getShelved now publish their default priority range (1 to 4). Diagnostics alarms (priority 0) are included only when minPriority is set to 0. alarm_get, alarm_count and alarm_getShelved no longer return Diagnostics alarms (priority 0) by default. Set minPriority to 0 to include them.
Changed	Configuration – Descriptions of tools, parameters, prompts and resources are now optional.
Changed	Resources – The tag and alarm resource templates are now named 'Tag' and 'Alarm'.
Changed	Help – Added help texts to every configuration group and improved the help texts for access, tokens, tools, prompts, resources, OAuth, method rules and ports.
Features	Protocol – Added support for the MCP 2026-07-28 stateless protocol revision. Clients using earlier revisions keep working from the same endpoint without sessions. Action required: the Sessions settings (Max concurrent sessions, Idle timeout) were removed, and resource subscriptions now require clients that support the 2026-07-28 revision. The Sessions settings (Max concurrent sessions, Idle timeout) were removed, and resource subscriptions now require clients that support the MCP 2026-07-28 revision.
Features	Tools – Custom tools can now be generated from a module method, inheriting its name, description and input and output schemas from the method picker.
Features	Logs – Added the log_read and log_stats tools to read module log entries filtered by time window, level, component and regular expression, and to count entries per module, day and level.
Features	Alarms – Added tools to shelve and unshelve alarms and to list shelved alarms.
Features	Tools – Added the node category (node_info, node_errorCount, node_getStandby, node_restart, node_generateNewUID), the authentication category for users, roles, local passwords and 2FA, and the system_getNetworkAdapters and system_getSerialPorts tools.
Features	Tags – tag_find now shows whether each tag is local or received through a link, and can filter tags by origin, quality, timestamp and value.
Features	OAuth – Added support for Client ID Metadata Documents (CIMD) client registration, the RFC 9207 iss parameter in authorization responses and the application_type field in Dynamic Client Registration.
Features	Tools – Added the 'Structured tool outputs' option (disabled by default): when enabled, custom tools publish their output schema and return structured content for programmatic MCP clients.
Features	Module Methods – Added the method_list, method_describe and method_call tools, which let MCP clients discover and call the methods published by module instances, enabled per token and filtered with include and exclude rules.

MCP Server (v.2.0.0)

Change	Description
Removed	Tools – Removed the system_status, system_errorCount, system_restartService and system_generateNewUID tools. Action required: update clients and prompts to use node_info and system_info instead of system_status, and node_errorCount, node_restart and node_generateNewUID for the others. The system_status, system_errorCount, system_restartService and system_generateNewUID tools no longer exist. Update clients and prompts to use node_info and system_info instead of system_status, and node_errorCount, node_restart and node_generateNewUID for the others.
Security	Access – Fixed an issue where tokens restricted by tag filters could list and read tags outside their allowed scope through certain tool arguments and tools.
Security	HTTP – MCP requests larger than 4 MiB are now refused with HTTP 413, tool calls with more than 10,000 argument elements are rejected, and JSON-RPC batches with more than 100 messages are refused, to prevent excessive memory usage. MCP requests larger than 4 MiB, tool calls with more than 10,000 argument elements and JSON-RPC batches with more than 100 messages are now refused, and clients using the 2026-07-28 revision must send the MCP-Protocol-Version header. Split large requests and update clients that do not send the header.
Fix	Access – Fixed an issue where tag filter rules denied access to tags that tag_find listed, because rule paths were case-sensitive and their regular expression also matched the rule's own path.
Fix	Resources – Fixed an issue where alarm resources returned alarms other than the requested one or failed with names containing regex characters, and where tag and alarm resources rejected unencoded paths.
Fix	Tags – Fixed an issue where tag_find failed when filtering by engineering units, history or source module and some tags did not define that field.
Fix	Server – Fixed an issue where the module restarted in a loop when its port was in use. The port is now retried every 10 seconds.
Fix	Configuration – Fixed an issue where an error occurred when opening a configuration whose tools, prompts or resources had never been saved.
Fix	Server – Fixed an issue where the module did not report its version to MCP clients during initialization.
Fix	HTTPS – Fixed an issue where the key path was saved as a password hash when the certificate type was 'File path', so the HTTPS server could not find the key file.
Fix	Resources – Fixed an issue where file resources were served corrupted because the data URI header was decoded as part of the file content.
Fix	Tools – Fixed an issue where module_getInstances did not filter by instance name and returned instances that matched only some of the filters, module_getConfigData did not report a configuration it could not find, and link_get reported a Store & Forward pause that was always false. Read tools now return null instead of Success when the node returns no data.

MQTT Client (v.1.16.0)

Change	Description
Changed	Publisher – A tag condition trigger with Initial change set to No now uses the tag's initial value as its starting point.
Changed	Logging – The module now logs a warning when a received message, a script output or a trigger tag is discarded, when the topics of two subscribers overlap, and when the broker closes a connection right after accepting it, which may indicate another client using the same Client ID. Repeated Store & Forward discards are summarized once an hour.
Changed	Help – Improved the help texts for Clean session, Session expiry interval, reconnections, the message interval, Cron and Regex, triggers, InfluxDB values, Custom script timing, 64-bit Protobuf integers, Retain handling and the default Store & Forward folder.
Features	Redundancy – The Store & Forward (S&F) backlog is now synchronized with the backup node, so pending messages are delivered after a failover. Only S&F folders and certificate files inside the node folder are synchronized, and a warning names any that are not.
Security	Authentication – The certificate and JWT private keys are now stored encrypted in the configuration and its backups instead of in plain form. Existing configurations keep working; select the key files again to encrypt the stored keys.
Security	Logging – The broker password, private keys, JWT tokens and MQTT 5 user properties are no longer written to the log in plain text, including in invalid configuration warnings and debug or trace output.
Security	Connection – The AWS IoT Core and Google IoT Core connection profiles now verify the broker certificate. Selecting them previously turned verification off. A warning is now logged at startup when the broker certificate is not verified.
Security	Subscriber – Limited the decompressed size of a received message to 64 MiB to prevent excessive memory usage.
Security	Dependencies – Updated the protobufjs, moment and ip-address libraries to versions without known vulnerabilities.
Fix	Persistent Session – Fixed an issue where QoS 1 and QoS 2 messages of a persistent session were duplicated or lost after a reconnection or a module restart. The session is now kept across restarts.
Fix	Store & Forward – Fixed an issue where messages already sent from the Store & Forward backlog were sent again after a restart, and where an interrupted disk write could deliver a truncated message. Action required: before downgrading MQTT Client to an earlier version, wait until the Store & Forward buffer is empty; earlier versions discard the pending data written by this version.
Fix	Connection – Fixed an issue where the module reconnected in a loop, or resent the same message indefinitely, when the broker closed the connection right after accepting it or after receiving a message. Messages larger than the MQTT 5 broker's Maximum Packet Size are now discarded before sending.
Fix	Connection – Fixed an issue where the connection was refused when the MQTT 5 broker did not support a publisher's Retain or QoS, a subscriber's QoS exceeded the broker maximum, or the broker did not support wildcard subscriptions.

MQTT Client (v.1.16.0)

Change	Description
Fix	Subscriber – Fixed an issue where shared subscriptions dropped every message, some topic filters matched topics of other subscribers, and a subscriber whose subscription the broker refused received messages of other subscriptions.
Fix	Subscriber – Fixed an issue where an older value could overwrite a newer one, tags sharing an address stopped updating, messages delivered on connection by a persistent session were discarded, and one invalid event discarded the whole update.
Fix	Parser – Fixed an issue where JSON messages starting with a UTF-8 BOM, compressed messages encoded in Base64 or Hex, and compact payloads of tags named like 'constructor' were not read.
Fix	Publisher – Fixed an issue where binary content published with the UTF8 encoding was corrupted.
Fix	InfluxDB – Fixed an issue where the InfluxDB line protocol output was invalid with null or NaN values or special characters in tag paths, and where apostrophes were removed from string values.
Fix	Publisher – Fixed an issue where a scheduled snapshot was merged into the next one instead of being skipped while the previous one was still being built, and where fixed-interval publishing did not start on the first interval boundary.
Fix	Tag Filters – Fixed an issue where tag filters and Tag change group triggers included sibling groups whose names start with the same text, and where group names with brackets or parentheses were filtered incorrectly or stopped the module.
Fix	Custom Scripts – Fixed an issue where a Custom script could alter the events of other publishers or receive null metadata when the tag model changed.
Fix	Publisher – Fixed an issue where a publish topic containing the wildcards + or # made the broker close the connection on every resend. The configuration now rejects these topics.
Fix	Configuration – Fixed an issue where the instance restarted in a loop when the configuration could not be read, JWT connections with additional claims failed to load, and an error writing Subscriptions.dat ended the process.
Fix	Redundancy – Fixed an issue where File path certificates and the Subscriptions.dat file were not synchronized with the backup node.

Notifier (v.1.5.3)

Change	Description
Fix	Mail Notifier – Fixed an issue where the custom Parameters of a mail notifier were lost when the configuration was reopened in the Web UI and saved.

OPC UA Server (v.1.18.0)

Change	Description
Changed	Certificates – Improved the error messages shown when the configured server certificate is not supported.
Features	Certificates – Added a certificate Type setting to choose between an Autogenerated self-signed certificate (default) and a Custom certificate, either self-signed or issued by a Certificate Authority (CA). Custom certificates and private keys can be uploaded or created with the certificate helper, which generates a self-signed certificate or a Certificate Signing Request (CSR). The private key is stored encrypted in the configuration.
Fix	Server – Fixed an issue where the module crashed when the OPC UA server failed to initialize, for example when the certificate uses an unsupported key type such as a multi-prime RSA key.
Fix	Historical Access – Fixed an issue where Raw HistoryRead requests never completed when numValuesPerNode was set to 0, and stalled when it was set to a very large value. A value of 0 now returns as many values per read as the server can provide, as defined by the OPC UA specification.
Fix	Historical Access – Fixed an issue where Historian errors during a Raw HistoryRead were not reported to the OPC UA client, and the request finished as if all values had been returned. The error is now returned with the corresponding status code.

REST API Server (v.1.10.0)

Change	Description
Changed	Subscriptions – Subscription events are now filtered by the user's tag permissions when they arrive, so the buffer limit and the discardedEvents and remainingEvents counters only include events the user can see.
Changed	Configuration – The module no longer restarts in a loop when the config.n3c or access.n3c configuration file cannot be read, for example when access.n3c does not exist. The error is now logged and the module stays idle.
Security	Authentication – Basic authentication no longer accepts a valid password followed by a colon and additional characters.
Security	API – Fixed an issue where a crafted query parameter in a GET request made every later request fail with an error until the module was restarted.
Security	Subscriptions – Fixed an issue where users or tokens without any included tags received the updates of every tag through subscriptions. They now get the same "no tags included" error as the other tag requests.
Security	API – Fixed an issue where a compressed (gzip or deflate) POST request with an invalid JSON body stopped the module. The module now answers with a 400 error.
Security	Subscriptions – Subscriptions can now only be listed, polled and cancelled by the user or token that created them. Other users get a "Subscription not found" error.

REST API Server (v.1.10.0)

Change	Description
Fix	Authentication – Fixed an issue where Basic authentication rejected valid credentials when the user name or password contained non-ASCII characters (such as ñ or ú) or the password contained a colon. Credentials are now decoded as UTF-8, as defined in RFC 7617.
Fix	Authentication – The 401 response now includes the realm in the WWW-Authenticate header (Basic realm="vNode REST API"), as required by RFC 7617.
Fix	Subscriptions – Fixed an issue where a subscription created while the last active one was being cancelled never received tag updates.

Siemens Client (v.1.8.5)

Change	Description
Changed	Configuration – The PLC IP address must now be a valid IPv4 address, and the Config Editor rejects addresses with zero-padded octets such as 192.168.001.010, which were read as octal and connected to a different PLC (192.168.1.8). Action required: remove leading zeros from PLC IP addresses in existing configurations. PLC IP addresses with zero-padded octets, such as 192.168.001.010, are now rejected as invalid and the device is not loaded. Remove the leading zeros (192.168.1.10) in existing configurations.
Fix	Connection – Fixed an issue where reads from online PLCs were delayed for minutes while several PLCs were unreachable, for example after a power outage. The module now checks TCP port 102 before connecting and no longer stacks reconnection attempts.
Fix	Devices – Fixed an issue where a single device with an invalid configuration, such as a rack or slot outside 0–255 or an invalid IP address, stopped the whole module. The device is now skipped with an error in the log, and its tags get BAD_CONFIG_ERROR quality.
Fix	Tags – Fixed an issue where tags that failed to load, for example because of an invalid rate, stayed as Bad [Uninitialized] without any log message. They now get BAD_CONFIG_ERROR quality and the reason is logged.

SNMP Client (v.1.3.3)

Change	Description
Removed	SNMPv3 – Removed the DES privacy protocol, which is insecure and failed on every request with "digital envelope routines::unsupported". Action required: change devices that use DES privacy to another privacy protocol. Devices configured with the DES privacy protocol are no longer created: they are flagged as invalid in the Config Editor and the module logs a warning. Change them to another privacy protocol.
Fix	MIB Files – Fixed an issue where MIB files whose file name contained a comma or a colon were not loaded correctly.

Sparkplug Client (v.1.4.0)

Change	Description
Changed	Primary Host – The log now shows the Sparkplug protocol version and the STATE topic the edge node is waiting for, making a version mismatch with the Primary Host easy to spot.
Changed	Help – Improved the help text of Check status to explain that the edge node waits for the first online STATE of the Primary Host, and which STATE topic and payload each protocol version expects.
Security	Authentication – The client certificate private key is now stored encrypted in the configuration and its backups instead of in plain form. Existing configurations keep working; select the key file again to encrypt the stored key.
Fix	Commands – Fixed an issue where NCMD and DCMD commands compressed with DEFLATE were discarded, so writes and rebirth requests from hosts using DEFLATE compression were ignored.
Fix	Connection – Fixed an issue where the client reconnected in a loop without delay when the broker closed the session right after accepting it, for example because of a duplicated Client ID. Reconnections now wait at least 5 seconds.
Fix	Devices – Fixed an issue where a device that no longer matched any tag after a tag model change kept publishing its previous tags with stale values in DBIRTH messages.

Sparkplug Host App (v.1.0.3)

Change	Description
Changed	Logging – A rejected malformed DataSet row is now logged with its row number, and the payload trace of births and data messages is only produced at trace level.

Sparkplug Host App (v.1.0.3)

Change	Description
Changed	Help – Improved the help texts for Compression and Host ID, explaining which STATE the host publishes and that Ignition MQTT Transmission cannot read compressed commands. The host app list is now labeled Host app.
Security	Payload – Updated the protobufjs library to version 7.6.6 to fix decoding vulnerabilities, and payloads with excessive nesting are now rejected instead of stopping the module.
Security	Templates – Fixed an issue where crafted template parameter or property names in a Sparkplug payload could alter internal objects of the module.
Security	Logging – The broker password and the client private key are no longer written to the log when a configuration is invalid.
Security	Authentication – The client certificate private key is now stored encrypted in the configuration and its backups instead of in plain form. Existing configurations keep working; select the key file again to encrypt the stored key.
Fix	Sessions – Fixed an issue where repeated, out-of-order or outdated messages from an edge node were applied, a birth with a sequence number other than 0 was discarded, and a late birth from a previous session triggered an unnecessary rebirth.
Fix	Sessions – Fixed an issue where retained messages and messages whose topic did not match their type were applied, devices missing from a new birth were not removed, and dead edge nodes were kept in memory indefinitely.
Fix	Sessions – Fixed an issue where rebirth requests were delayed or repeated too often. An edge node already publishing when the host started no longer waits for the Reorder Timeout, and device rebirth requests now name the Device Control/Rebirth metric.
Fix	Connections – Fixed an issue where the host reconnected in a loop after a broker dropped the session, a stopped host app reconnected on its own, and losing one of several brokers did not degrade the edge nodes reached only through it. The online STATE and rebirth requests are now sent again after reconnecting.
Fix	Connections – Fixed an issue where a certificate kept from another authentication mode turned an MQTT connection into TLS, and where a certificate or key that was not selected as a file in the editor was not reported.
Fix	Tags – Fixed incorrect tag qualities. Missing array elements, DataSet cells and template instance tags no longer show null with good quality. Tags now show BAD_COMM_FAILURE when no broker is reachable and BAD_CONFIG_ERROR when the host app is disabled. Tags of a dead device take the DDEATH timestamp.
Fix	Metrics – Fixed an issue where several samples of a metric in one data message reached the tag as the last value repeated, and older historical samples left the tag on an old value. Metrics without their own timestamp now take the payload timestamp.
Fix	Metrics – The Quality property is now read from any integer type and from Ignition quality codes (Uncertain is UNCERTAIN; Bad and Error are BAD). Unknown codes give UNCERTAIN, and data without quality is GOOD.
Fix	Templates – Fixed several alias and template issues: metrics without an alias took alias 0, aliases above 2^53 collided, template definitions were shared between edge nodes, updating a parameter of one instance changed the definition and the other instances, and definitions sent in a DBIRTH were only partially loaded.

Sparkplug Host App (v.1.0.3)

Change	Description
Fix	Data Types – Fixed several decoding issues: empty strings were read as null, empty elements of string arrays were dropped, zero timestamps were rejected, DateTime values before 1970 were read incorrectly, DateTime DataSet cells did not reach their tags as epoch milliseconds or ISO text, and a null DataSet cell rejected the whole node.
Fix	Writes – Fixed several write issues: values are now converted exactly or the write is refused, null is sent as is_null, commands keep the BIRTH data type, nested template writes keep every intermediate level, DataSet, Bytes and File writes keep their original representation, and writes now fail when no broker can publish the command.
Fix	Browser – Fixed an issue where a tag created from the browser for a device metric was not bound to that device, and a missing device was not reported as 'Device not found'.
Fix	Tag Model – Fixed an issue where an error in the tag model subscription stopped the module.

SQL Client (v.2.0.0)

Change	Description
Changed	Connection Pool – Each channel now uses a pool of connections and runs its transactions concurrently, up to Max pool connections, and triggers fire as soon as their condition is met. Order is kept within a transaction. Action required: set Max pool connections to 1 if the transactions of a channel must not overlap. Transactions of the same channel can now run at the same time, up to Max pool connections, instead of one after another, and triggers no longer wait for the previous transaction to finish. If your transactions must run in sequence, set Max pool connections to 1 in the channel.
Changed	Transactions – Retry attempts now applies to Standard transactions only and retries only transient failures, with exponential backoff. History inserts are retried by Store & Forward.
Changed	Tags – When a Standard transaction fails, only its own tags get BAD_DEVICE_FAILURE quality, through a quality-only update that keeps their last value. The module no longer goes offline when the database is unreachable.
Changed	Scripts – $.parameter, $.event and $.logger are now read-only in custom scripts. Action required: update scripts that assign to $.parameter or $.event, or to anything inside them; copy the value into a variable instead, and keep in $.local what must outlive an execution. Custom scripts (the query, parser and serializer of a transaction, the scripts of a query method, and the scripts run from the Config Editor buttons) that assign to $.parameter or $.event, or to anything inside them, now fail with a TypeError instead of changing them for the next executions. Copy the value into a variable, for example const v = $.parameter.x.value * 10, and keep in $.local what must outlive an execution.

SQL Client (v.2.0.0)

Change	Description
Changed	Drivers – Updated the SQL Server, Oracle, MySQL and PostgreSQL database drivers. Action required: SQL Server connections now require TLS 1.2 or later, so very old unpatched servers may refuse to connect; Azure AD password authentication now requires a Client ID. SQL Server connections now require TLS 1.2 or later, so servers that only support older TLS versions refuse the connection; update or patch them before upgrading. Azure AD password authentication now requires a Client ID; add it to the connector configuration.
Changed	Configuration – Connector Parameters strings are now parsed as URL query parameters: values may contain '=', %xx escapes are decoded, and '+' is read as a space. Action required: replace '+' with %2B in parameter values that need a literal plus sign. Connector Parameters are now parsed as URL query parameters, so a '+' in a value is read as a space and %xx sequences are decoded. Replace '+' with %2B in values that need a literal plus sign.
Changed	Help – Added help texts to every configuration group and improved the help texts for Store & Forward, Retry attempts, Drain delay, Statement timeout, SSL paths and the scripts. The default Serializer script includes, commented out, the line that skips initial events.
Features	Store & Forward – History transactions can now buffer their events on disk with Store & Forward, so the backlog survives module restarts and long database outages. Failed batches are retried until the database recovers, and events older than Max. days in disk are discarded.
Features	Query Methods – Added query methods: named queries published as call methods in their own Query methods node. Each one chooses its channel and declares its query, parser, input and output schemas, and can be tested with a Run popup that always rolls back.
Features	Methods – Configured transactions are now published as call methods (execute and dryRun). The module also publishes lab tools to test a connection, explore schemas and tables, and run a statement that is rolled back by default.
Features	SqlClient Lab – Added the SqlClient Lab page to list the schemas and tables of a saved channel, describe a table and preview its rows.
Features	Config Editor – Added a Test connection button to the Connector group, which tests the values in the editor before saving. Added a Run button to transactions, which runs the query, parser and serializer of the editor without writing tags.
Features	Scripts – Query scripts can now return a bound statement { text, values } with :name placeholders. The values are passed to the database driver separately from the SQL text instead of being interpolated.
Features	Scripts – Custom scripts now receive $.event.ts, the timestamp of the trigger fire that started the execution, and query scripts can use Buffer to build binary values.
Features	Configuration – Added the Max pool connections setting, the Statement timeout for PostgreSQL and the Request timeout for SQL Server. For SQL Server, 0 means no limit and replaces the driver's previous implicit 15-second limit.

SQL Client (v.2.0.0)

Change	Description
Removed	Configuration – Removed the channel Timeout and Inter-request delay, the transaction Auto-demotion and the connector Reconnect delay options; existing values are ignored and logged at start. Dead connections are now detected in about 35 seconds and unresponsive servers in about 70 seconds; use the database statement timeouts to limit slow queries. The channel Timeout and Inter-request delay, the transaction Auto-demotion and the connector Reconnect delay settings are ignored after the upgrade. Use the database statement timeouts to limit slow queries.
Security	SSL – The SSL private key is now stored encrypted in the configuration and its backups instead of in plain form. Existing configurations keep working; select the key file again to encrypt the stored key.
Security	Authentication – The Azure AD client secret and access token are now stored encrypted in the configuration.
Fix	Triggers – Fixed an issue where Cron triggers fired a few milliseconds early, causing duplicate-key row losses. Also fixed: Fixed time triggers realigned their cadence after a reset, monthly or yearly schedules fired immediately, and null or not-null tag conditions ignored their On false mode.
Fix	Transactions – Fixed an issue where a transaction stopped mid-execution still published its result or wrote its reset value, a parse failure ran the SQL statement again, and a failed History insert lost its events. Also fixed: duplicate subscriptions inserted duplicate rows, and group parameters such as Area/1 also matched Area/10.
Fix	Data Types – Fixed several data conversion issues: MySQL binary columns arrived as text, Oracle LONG columns were read as integers, empty MySQL results lost their columns, unknown column types and duplicated column names failed the query, NaN and Infinity were written as wrong values on MySQL, SQL Server and Oracle, and zero timestamps corrupted the tag history.
Fix	Connections – Fixed an issue where encrypted SQL Server connections to an IP address failed, and a PostgreSQL connection lost during a statement stopped the module. Also fixed: an invalid channel, transaction or query method stopped the module in a loop, and a tag model subscription error stopped the module. The Azure AD tenant ‘customers’ is migrated to ‘consumers’, and a MySQL port of 0 now uses 3306.
Fix	Scripts – Fixed an issue where a script with an infinite loop froze the whole module; scripts are now stopped after a script timeout (30 seconds by default). Also fixed: legacy scripts failed to compile, and a parameter named __proto__ was lost.
Fix	Redundancy – Fixed an issue where the File path SSL certificates of a channel were not synchronized with the backup node, so the channel did not start there.

Web UI (v.1.26.0)

Change	Description
Changed	Certificates – The certificate generation popup now fills the hosts with the host name, localhost and the IP addresses of the network interfaces, and the 'Host name' field is now labeled 'Common name'.
Changed	Config Editor – Saving now renews the configuration lock, so when a save fails the user keeps the full lock timeout to fix it without losing the unsaved changes.
Changed	Config Editor – When a configuration is too large to be saved, the Config Editor now shows its size and the maximum allowed, recommends using templates to reduce it, and keeps the unsaved changes.
Changed	WebSocket – Increased the maximum WebSocket message size from 200 MB to 400 MB, so larger configurations can be saved.
Changed	Config Editor – Encrypted file pickers now show '<encrypted>' instead of '<embedded>' as their output.
Changed	Historical – Each user can now save up to 100 configurations of each type, of up to 512 KB each. Overwriting an existing configuration is still allowed when the limit is reached.
Changed	Config Editor – Instances with nothing to show no longer display an expand arrow.
Features	Certificates – The self-signed certificate and Certificate Signing Request (CSR) generation now accepts custom SANs and key usage options, and modules can prefill the certificate fields before generating it.
Security	Historical – Enforced server-side checks so users can only load their own private Historical configurations.
Fix	Certificates – Fixed an issue where the certificate generator added every host to the Subject Alternative Name (SAN) both as an IP address and as a DNS name. Each host is now added with the type it matches.
Fix	Config Editor – Fixed an issue where setOptions or updateOptions on a library template with instances failed and left the template and its instances partially updated.
Fix	WebSocket – Fixed a memory leak where closed browser connections were kept in memory until the module was restarted.
Fix	Config Editor – Fixed an issue where file pickers that store the file name could not download the file or show its name when the name contained commas, semicolons or colons. These characters are now stored as underscores.
Fix	Historical – Fixed an issue where users with View permission got a permissions error when saving or deleting their private presets or changing 'Load on login'. Saving public presets still requires Edit permission.
Fix	Historical – Fixed an issue where Historical configurations synchronized by redundancy were not listed, loaded or deleted until Web UI was restarted.
Fix	Historical – Fixed an issue where Web UI failed to start when a configuration from an older version could not be moved to the new data folder. The file is now kept with a warning and moved on a later start.

Web UI (v.1.26.0)

Change	Description
Fix	Historical – Fixed an issue where deleting a configuration reported success even when it failed, and a copy in the other public or private folder reappeared.
Fix	Config Editor – Fixed an issue where templates with an empty result failed to load and their instances could not be saved.
Fix	Config Editor – Fixed an issue where a template with an invalid result was loaded anyway after its error was reported.

Web Vision (v.1.13.0)

Change	Description
Changed	Performance – Components of templates that are not subscribed no longer poll, redraw or recalculate properties. The Properties Tree and the designer panels redraw at most once per frame.
Changed	Libraries – Updated Highcharts to version 12.6.2 and ag-grid to version 27.3.0.
Features	Designer – Redesigned the designer layout. The new Outline tree is on the left and the Properties on the right, both with search. The Value and Output columns can be resized, with clipped values shown in a tooltip, and the titlebar buttons have tooltips.
Features	Designer – Shift+click now selects the innermost component under the cursor, and each repeated Shift+click selects its parent. The read-only Value cell of an inherited property jumps to the component that declares it.
Features	Instance Lists – Instance lists can now build their children, series, datasets and columns from a Data expression.
Features	Color Picker – Added a color picker with palette, recently used colors and HSV editor to color properties.
Features	Actions – The actions of an event can now be exported to and imported from a CSV file.
Features	Dropdown – The Dropdown component accepts multi-level Options from an expression that returns a nested object, and custom dropdown properties support nested groups.
Features	Chart – Added the Download chart CSV event action, a per-series line style option (such as dashed lines) and the spline series type.
Features	Shapes – Added a dash pattern option to the strokes of the Ellipse, Line and Rectangle components.
Features	Tables – Added an Expand column width mode that fills the remaining table width. A column width set by dragging is kept until its separator is double-clicked.
Features	Subscription – Added the Unparented subscription setting to root templates and popups, so they subscribe to their tags only when they are shown.

Web Vision (v.1.13.0)

Change	Description
Features	Charts – The Heatmap exposes the Y axis text color and the gradient legend label color and font size, and the Pie Chart exposes the data label font family.
Security	Distributed – Fixed an issue where an unauthenticated client could make the server expand a WebSocket header to a very large size. The remote node is now authenticated first, and the expanded size is limited.
Security	Reports – PDF reports now accept only PNG and JPEG images, so report images can no longer read other files from the server, and the error text of a failing chart is escaped.
Fix	Templates – Fixed an issue where the Designer and the Viewer stopped responding or never finished loading the project when a property, a child or a template extension referred to a missing template. A property set to a value outside its list now logs a warning with the allowed values.
Fix	Actions – Fixed an issue where a try/catch did not catch a failed read of a missing tag and showed an error popup, and where a finally block replaced the value returned by try or catch in expressions.
Fix	Touch – Fixed an issue where context menus, long popups, the login form and scrollable chart legends could not be scrolled with a touch drag on phones and tablets.
Fix	Sound – Fixed an issue where a muted sound showed an error popup when Chrome paused a background tab.
Fix	Web Server – Fixed an issue where a request with an undecodable URL stayed open until the server timeout. It is now answered with a 400 error.
Fix	Config Editor – Fixed an issue where the property picker did not offer the sub-template variant properties, for example the shelving properties of the Alarm Table.
Fix	Tooltips – Fixed an issue where tooltips and context menus stayed invisible or off screen when they needed repositioning, and a tooltip appeared after being closed during its delay.
Fix	Tables – Fixed an issue where autofit columns were not remeasured after an inline cell edit, and confirming an IME composition ended the cell edit.
Fix	Templates – Fixed an issue where an error in one callback stopped a whole polling group or the propagation of outputs, and the Tag Picker showed an outdated browse response.
Fix	Dropdown – Fixed an issue where cancelling the change confirmation did not restore the previously shown option.
Fix	Files – Fixed an issue where images, sounds, report fonts, downloads, the login logo and PDF images picked by file name were not loaded.
Fix	Popup Windows – Fixed an issue where copying to the clipboard, tooltips, text selection and cell editing in a popup window acted on the main window instead.
Fix	Inputs – Fixed several text editing issues: a committed value was shown scrolled to the caret, Escape wrote the previous value back when the cell lost the focus, the write popup did not select the tag as soon as a value was typed, and confirming an IME composition closed the Properties Tree search.

Web Vision (v.1.13.0)

Change	Description
Fix	Actions – Fixed an issue where Set busy blocked all events of the component instead of only the busy event, and a runtime input stayed editable while its User input actions were busy.
Fix	Actions – Fixed an issue where modules(), links() and nodeStatus() failed with a TypeError in event actions.
Fix	Layout – Fixed several layout issues: children sized in percent did not resize with their layout, children after one with an empty Size were pushed out of the layout, a padded child did not fit inside a Region, and scaled padded components moved when their anchor was dragged.
Fix	Tables – Fixed scroll issues in long tables: the scroll handle was not drawn when the row height was not set, it did not resize when the data changed, and the wheel and the handle drag did not reach the last row.
Fix	Alarm Table – Fixed an issue where pressing Enter in the acknowledge message did not acknowledge the alarm.
Fix	Reports – Fixed an issue where the PDF header colors and the hide reload and download options could be edited in Designer mode.
Fix	Pie Chart – Fixed a stack overflow when a style change redrew slices of different values.



vNode

info@vnodeautomation.com
sales@vnodeautomation.com

USA: (+1) 754 755 0009 **UK:** (+44) 161 660 32416 **SPAIN:** (+34) 935 721 007 **COSTA RICA:** (+506) 222 523 44
FRANCE: (+33) 041 368 0106 **MEXICO:** (+52) 554 628 2593

4855 W Hillsboro Blvd STE B3, Coconut Creek FL 33073, USA