Update 5c - Detailed Release Notes

Complete feature and improvement details for FrameworX 10.1 Update 5c, organized by product area.

FrameworX 10.1 Update 5c — June 22, 2026

Build: 2056

OPC UA

- **Structured-type UDT tags over ExtensionObject now decode correctly** — OPC UA devices that expose UDT tags as ExtensionObject — including CODESYS and WAGO PLCs — now read all struct members with accurate values at runtime. Nested structures are fully supported.
- **Data Explorer drills into nested struct nodes** — the OPC UA Data Explorer now expands struct hierarchies beyond the root level, making it straightforward to inspect and map deeply nested UDT members from any OPC UA server.
- **TagProvider browses nested struct members** — the OPC UA TagProvider now exposes sub-members of nested struct nodes, completing full struct-hierarchy visibility from tag browsing through to runtime.
- **OPC UA Simulator enforces configured security policies** — the built-in OPC UA Simulator now applies exactly the security policies configured, with no silent fallback to unsecured endpoints.
- **Address space commas preserved** — node identifiers containing commas in the OPC UA address space are now stored and used correctly in Devices configuration.
- **CertificateType — Custom option available** — the CertificateType field on OPC UA devices now includes the “Custom” option, enabling managed certificate workflows for deployments that require it.
- **Custom certificate trust — no repeat prompts** — once a server certificate is trusted with the Custom policy, subsequent connections proceed without re-prompting. The certificate trust prompt in the import wizard is also shown only once.
- **Tag import resilience** — OPC UA tag import completes successfully even when the server is intermittently unreachable or the configuration references a legacy format.

AI Integration — MCP and AI Authoring

- **Headless solution execution for automated testing** — AI agents and CI pipelines can now start a solution headlessly with its full web client available, enabling end-to-end automated testing without a visible desktop.
- **Device tag-import wizard automation** — import wizards and configuration-grid checks for device tags are now fully drivable from AI tools and automated scripts.
- **Partially-configured rows from automation** — AI agents can create mid-commissioning (partially configured) rows explicitly, supporting incremental provisioning workflows.
- **Display clipboard across solutions** — AI Designer can now copy and paste Display content between different solutions using new clipboard export and import tools.
- **Author display configuration completely from automation** — legend, connections, chart orientation, child-display links, and themes can all be authored from automation tools without opening the Designer UI.
- **Verify dialog content in automated tests** — dialog content and prompts in automated Designer tests can be inspected without rendering them, enabling headless CI validation of dialog-driven workflows.
- **Validate electronic-signature workflows from automation** — electronic-signature and security workflows can be exercised safely through automated tests, with full audit-trail fidelity.
- **Local AI — tag property read coverage** — the Local AI Chat can now read all tag properties, including type, reference tag, and startup value.
- **LocalAI shipping configuration** — the SolutionCapabilities dialog and ModelOptions defaults now reflect the Local AI shipping configuration, ensuring out-of-the-box behavior matches documentation.
- **UNS Asset Tree folders — Description field** — asset folder descriptions can now be set and updated through AI tools, completing the metadata round-trip for UNS assets.
- **Rendered color reading** — AI tools can now read the actual rendered color of display elements at runtime, enabling color-aware automation and visual validation.
- **Imported solutions open their own main page** — solutions opened via AI tools now navigate to their own main page; long builds report progress rather than timing out silently.
- **Browser auto-launch on web-client solutions** — starting a .NET 10 solution that includes the web client now automatically opens the browser, matching the interactive Designer experience.
- **Write tools report accurate completion status** — MCP write tools no longer return success when the Designer was busy or disconnected at the time of the call.
- **Misspelled field warning on writes** — display and script writes that include an unrecognised field name now surface a warning, preventing silent data loss from configuration typos.
- **Query and script content integrity check** — saved queries and script content are verified against what was sent; a notice is raised when the editor internally overwrote the intended value.
- **Dataset query SQL integrity** — dataset queries created or edited through AI tools now reliably save exactly the SQL that was provided. JSON workspace imports no longer corrupt tag references inside SQL text.
- **ConsoleMCP — element type validation** — write_objects rejects unknown display element Type strings; verify_workspace flags elements whose Type is not in the element catalog, preventing silent misconfiguration.
- **ConsoleMCP — protocol and element schema completeness** — list_protocols now returns Point Address schemas for DataHubService and LabView; list_elements eliminates duplicate Polygon/Path shape-alias entries.
- **ConsoleMCP — workspace isolation** — workspace stop and auto-close actions target only their own runtime instance, never affecting other open workspaces.
- **ConsoleMCP — trailing backslash fix** — create_solution_file no longer hangs when the output path ends with a trailing backslash.
- **AI Runtime connector — local runtime discovery** — the AI Runtime connector reliably locates locally running runtimes across all configurations.

- **AI Chat — stability across long sessions** — the AI Chat control remains responsive and leak-free across display reloads and extended operator sessions.
- **User-account fields in AI tools** — security user fields such as Blocked are now fully accepted by the AI authoring tools, completing the security-user schema.
- **Display Description persists on update** — a display's Description field is now correctly saved when an existing display is updated through the AI Designer.
- **verify_solution_file — Script Expression inventory** — the tool now inventories Script Expressions and correctly finds solutions stored in non-default folder locations.
- **Element round-trip fidelity** — re-writing a normalized display through ConsoleMCP now correctly reports a change, enabling round-trip diffing and idempotency checks.
- **Responsive canvas — element-lock behavior documented** — element-lock behavior in the AI Designer context and Display schema is documented, ensuring consistent automation results on responsive-canvas displays.
- **AI Chat — Secure Gateway routing** — the Chat panel shows replies correctly when LLM traffic is routed through a Secure Gateway egress server, supporting air-gapped and cloud-hybrid deployments.

Visualization and Displays

- **Symbol Legend** — symbols with *Has Legend* enabled now render their legend correctly; the value shown in the legend is fully configurable per symbol.
- **Per-instance enumeration override** — a tag's enumeration can be overridden on a per-instance basis, giving display authors precise control over how enumerated values appear on individual symbol instances.
- **Operator range-validation warning** — when an operator enters a value outside a tag's configured min/max range, a clear warning is displayed before the value is applied.
- **Pinch-to-zoom on mobile and mobile browser** — pinch-to-zoom gesture now works correctly on displays accessed from the mobile app or a mobile web browser.
- **Connections within symbols** — Allow Connections now correctly applies to shapes inside a symbol, enabling connector-based graphics inside symbol definitions.
- **Runtime theme propagation** — themes edited at runtime now update correctly across all open displays, not only the display where the change was applied.
- **Trend Chart — Entire Page expand button (vertical orientation)** — the Entire Page expand button is now visible and functional when the chart orientation is set to vertical.
- **Chart expand buttons and dashboard maximize reliability** — chart expand buttons and the dashboard maximize action continue to work correctly after re-binds and repeated interactions.
- **Knowledge Graph — reopen stability** — the Knowledge Graph display remains healthy across repeated reopen cycles and no longer blocks screenshot capture.
- **Knowledge Graph — stale binding cleanup** — tag-binding registrations accumulated from previous open cycles are released correctly when the display is reopened.
- **Display cleanup — gauge, theme, read fidelity, PNG capture** — linear gauge scale reference after live interval, Knowledge Graph theme handler, full-detail display read fidelity, and PNG capture scaling are all resolved.
- **Popup dialogs above embedded web browser content** — popup dialogs no longer open hidden behind embedded web-browser content on displays.
- **Text box value application** — text box input values are now applied correctly without requiring focus to move to an object with an action.
- **Radial Gauge — value indicator clipping** — the value indicator on the TRadialGauge no longer overflows the gauge frame at extreme values.
- **Trend Chart and Drilling Chart — Entire Page expand button** — the navigation-bar Entire Page button renders its icon and responds correctly on both chart types.
- **Dashboard panel maximize reachability** — the maximize button on dashboard panels remains accessible when moving the cursor toward it.
- **MainPage protection** — the solution's home page (MainPage) is protected from accidental rename or delete on all edit paths in the Designer.
- **Screen navigation responsiveness** — screen navigation no longer stalls or becomes unresponsive to clicks while the current screen still has objects loading.

HTML5 Web Client

- **Display buttons remain responsive on custom ports** — display action buttons in the HTML5 client no longer become unresponsive after navigating away and back when the server is configured on a port other than 3101.
- **Vector shapes with Path geometry now render** — display shapes authored with Path geometry (valve bodies, connector arrows, and similar) now render correctly in the HTML5 web client.
- **Drilling Chart renders in HTML5** — Drilling Chart displays render correctly in HTML5 web pages across all configurations.
- **Chart and alarm toolbar icons** — toolbar icons for chart controls and AlarmViewer display correctly in HTML5 web pages across all themes.
- **Electronic-signature cancel — HTML5** — canceling the e-sign dialog in the HTML5 client no longer executes the protected action or applies the protected value change.

Security

- **On-Screen Keyboard — password masking** — the on-screen keyboard preview no longer exposes password characters in plain text.
- **Security Secrets — field visibility for authorized users** — the Secret value field is now visible to every user with permission to edit Security, not only to the Administrator account.
- **Electronic-signature cancel — clean close** — canceling an e-sign dialog fully closes it with no lingering timer and no stray audit entries.
- **Electronic-signature workflow automation** — electronic-signature and security workflows can be validated through automated tests with full audit-trail fidelity.
- **Solution Center licensing calls resilience** — Solution Center licensing calls now reuse connections, recover after network loss, and handle forward-slash paths correctly, reducing activation failures in complex network topologies.
- **Windows service path hardening** — service registration for TStartupService, TWebServices, MQTTBroker, and Hardkey now uses quoted executable paths, eliminating the unquoted service path vulnerability.

Designer

- **Symbols imported with complete graphics** — symbols imported from the library or from XAML files are saved with their full graphical content intact. Symbol loading continues to work correctly after import.
- **Visual Query Builder — multi-language support** — the embedded Visual Query Builder loads the database schema tree correctly regardless of the active Designer UI language.
- **Ontology export/import round-trip** — ontology names survive an export/import cycle without modification; malformed ontology files are rejected safely without corrupting the solution.
- **Read-only plugin — apparent delete prevention** — Alarm Areas and UNS asset folders from a read-only plugin no longer appear to delete or rename successfully; the Designer correctly reflects that changes are not persisted.
- **Drawing tool cancel** — canceling a drawing tool mid-draw no longer leaves leftover connections or phantom undo entries in the canvas.
- **Security Secrets field visibility** — see Security section above.
- **EngWrapper — enumeration support** — enumeration values can now be defined through the EngWrapper API; creating user templates and enumerations with members works reliably.
- **EngWrapper — InsertDisplay with LF line endings** — InsertDisplay no longer silently ignores element lists that use LF line endings, ensuring cross-platform automation scripts produce correct results.
- **EngWrapper — GetDisplaySymbols and RemoveAllSymbols** — API errors in GetDisplaySymbols and RemoveAllSymbols are resolved.
- **EngWrapper — CSV import of custom profile settings** — CSV import of custom profile settings now imports all entries correctly.
- **Designer editor reliability** — multiple fixes to the display editor: export thread safety, clipboard bracket handling, lock checks, build flag consistency, list refresh, and child-display reopen behavior.

Runtime and Solution Merging

- **Solution Merging — improved detection and completeness** — secondary solutions named `Main__Module__.dbsln` are now correctly detected; Access Types defined in secondary solutions are merged into the running solution.
- **Runtime health endpoint — execution profile** — the runtime health endpoint now reports the active execution profile, giving operations teams richer visibility into runtime configuration state.
- **Network error recovery** — runtime communications recover cleanly from transient network accept errors without requiring a restart.
- **Async library compatibility** — all runtime executables load the updated async library correctly after the dependency upgrade, eliminating startup failures on some environments.
- **Script Class method calls on Linux runtime** — calling a Script Class method from a Display CodeBehind no longer times out on the Linux runtime.
- **Configurable log file rotation** — log file size limits and daily-archive retention are now configurable per runtime process, supporting long-running unattended deployments without unbounded disk growth.

Devices and Protocols

- **PEER Protocol / Microlok II — new driver** — native driver support for PEER Protocol-based devices, including Microlok II controllers used in rail and transit automation.
- **SiemensPlus — S7-1500 string handling** — S7-1500 string tag handling is reviewed and corrected, ensuring reliable read/write of STRING types across all S7-1500 configurations.
- **TwinCAT — recovery after PLC restart** — the TwinCAT driver now recovers correctly after a PLC restart, eliminating the ADS 1809 reconnection loop that required a runtime restart to resolve.
- **Docker bridge mode — redundancy parameters** — redundancy parameters are now correctly passed to TStartup in Docker bridge mode, enabling proper redundancy operation in containerized deployments.
- **Runtime symbol propagation** — symbols edited at runtime now update correctly on all open displays, not only on the display where the edit was made.
- **Cross-project Display asset copy** — copying Display objects across projects now correctly carries all assets, including symbols and images for both grouped and ungrouped selections.
- **Solution Center Concurrent Users count** — the Concurrent Users count in Solution Center is now sourced correctly; Rich Client Viewer and Viewer licenses are displayed separately.
- **Export Plugin selection** — the “Create a Plugin” action in the Export Plugin no longer auto-selects unintended solution elements.
- **AlarmViewer Category column** — the Category column in AlarmViewer and Property Watch is now populated correctly at runtime.
- **Symbol rectangle color stability** — symbol rectangle colors no longer revert to red after the symbol is saved.

Reports

- **Report Server startup resilience** — the Report Server starts cleanly even when the runtime server is not reachable at boot time; connectivity is retried in the background.
- **Report regeneration after edit** — regenerating a report after editing it no longer returns an “invalid content” error; the Trace Window and Last Execute Status reflect the correct result.

FrameworX 10.1 Update 5b — Released June 5, 2026

Resolved from Update 5

- **Visual Query Editor** — the schema tree now reflects the actual tables and columns from the connected Dataset database, enabling live schema-driven query design. The embedded Visual Query Editor is now fully operational.
- **Industrial Ontology import** — JSON-LD (`.jsonld`) and Turtle (`.ttl`) formats are now importable, completing import/export format symmetry: all four formats (RDF/JSON, N-Triples, JSON-LD, Turtle) are available in both directions.

New Features and Enhancements

- **OIDC / OAuth2 — SSO Sign-In Button:** The SSO entry-point button on the client logon surface is now active. Operators see a direct single sign-on option alongside the standard login; identity provider sessions are managed transparently through the configured Keycloak, Auth0, Entra ID, or login.gov provider.
- **AI Chat — Multi-Provider and Richer Conversations:** AI Chat now supports multiple simultaneous providers — Local AI, Anthropic Claude, OpenAI GPT-4o, Google Gemini, and Azure OpenAI — with the active provider selectable per conversation. The conversation experience adds streaming responses, code-block rendering with syntax highlighting, and one-click copy, making AI-assisted operator workflows more productive.
- **Local AI — Simplified Setup and SecureGateway Routing:** A guided first-run wizard streamlines Local AI service configuration. SecureGateway routing enables AI endpoints hosted in the cloud to be accessed through the FrameworX gateway, simplifying hybrid and air-gapped deployments.
- **Runtime REST API — Tag Writes and Standardised Errors:** The Runtime REST API gains tag-write support via `PUT /api/v1/uns/group` (one or more tag values per request), complementing the existing read and history capabilities. All REST endpoints now return RFC 9457 Problem Detail error responses, providing consistent, machine-parseable error handling for IT integrations.
- **SiemensPlus — Cross-Platform Siemens Driver:** Update 5b introduces SiemensPlus, a new Siemens driver built on a modern API foundation with native cross-platform support — running on Windows, Linux, and Docker-hosted FrameworX deployments. Existing solutions using the original Siemens driver continue to operate without any changes; the legacy driver remains fully supported for backward compatibility.
- **Combo Box and List Box — Live Enumeration Source:** Combo Box and List Box controls can now source their data items directly from a Tag enumeration definition. Values and display labels update dynamically as the enumeration changes — keeping dropdowns in sync with process configuration without manual maintenance.
- **Linear Gauge — Fully Dynamic:** Fill colour, minimum, maximum, and all range-band boundaries on the Linear Gauge are now bindable to tags. Gauge appearance responds live to runtime value changes, enabling process-aware visualisations without scripting.
- **Security Secrets — At-a-Glance Credential Health:** The Security Secrets dialog is redesigned around operational clarity: each secret shows its current status (Valid / Invalid / Expired), expiry date, and a *Rotate* action — giving administrators a single, actionable view of credential health.
- **Industrial Ontology — Faster Large-Scale Imports:** Large ontology imports are significantly faster, with reduced memory usage during RDF graph traversal. Deployments with complex or extensive ontology definitions see meaningful performance improvements on import and reload.
- **Device Nodes — Category Field:** A new *Category* field on Device Nodes enables grouping and filtering in the Device Explorer tree, improving navigation and organisation in solutions with large numbers of devices or complex architectures.
- **Solution Center — Offline License Activation:** Air-gapped deployments can now activate FrameworX licenses without internet connectivity. A QR-based activation code exchange allows a connected machine to generate and transfer the activation to the isolated target machine.
- **JSON Import/Export — More Control:** Exports support filtering by tag type, enabling targeted round-trips for specific asset categories. Import correctly handles circular UserType reference structures that previously required workarounds.

Improvements

OPC UA

- Remote engineering certificate management is more robust: the private key is correctly preserved across saves, and the Designer certificate list remains clean.
- Server endpoint URLs are validated before a connection is attempted; feedback is surfaced immediately in the TraceWindow.
- Reconnection after a server restart completes within the configured timeout.
- Subscription recovery after a lost connection correctly reflects only actual value changes, preventing stale data from appearing as live updates.

ControlLogix

- Program-scope tags are read and written correctly when the controller is in Remote Program mode.

Bailey Driver

- Full communication is maintained for solutions upgraded from FactoryStudio 9.x.

TraceWindow

- The module name column is auto-sized on first open; wide module names are accommodated without layout impact.

Displays

- `Shape.StrokeThickness` changes applied via script take effect immediately.
- Portable controls (`TRadialGauge`, `TKnowledgeGraph`) inside a `TChildDisplay` loaded from a different solution render correctly in all configurations.

Trend Chart

- Real-time data for a newly added pen appears immediately, without requiring a manual time-range change.

Reports

- Page breaks in multi-page tables are positioned correctly in PDF output, eliminating any row overlap.
- Embedded images in HTML export are Base64-encoded inline, producing fully self-contained HTML files.
- The Alarm Linear Gauge report element renders correctly for all alarm count values.

Designer

- Theme changes (Light/Dark) propagate to all open property editors without a Designer restart.
- The *Clone from template* step in the Solution Wizard handles paths that contain non-ASCII characters.

Runtime

- The .NET 10 runtime correctly binds to ports 80 and 443 when running as a Windows Service.
- SQL Server connections default to `Encrypt=Optional` for compatibility with older on-premises instances; the `TrustServerCertificate` flag is exposed in the connection string helper.

Alarms

- The alarm count badge on the AlarmViewer toolbar reflects acknowledgements made via the REST API immediately.

Scripts

- E-Sign triggers defined in a Script Task's `OnValidated` event fire correctly when the task runs from an automation schedule.

FrameworX 10.1 Update 5 — Released May 26, 2026

Platform Foundation

.NET 10 (LTS) Runtime
FrameworX 10.1 Update 5 ships on .NET 10 Long-Term Support — Microsoft's latest and most performant runtime baseline. Every server deployment, Web Contents target, and Designer Web host benefits automatically. The upgrade is handled entirely by the installer; no solution changes or manual steps are required. .NET Framework 4.8 continues to be supported for environments where it is needed.

Within-Major Compatibility
Every 10.1.x solution opens and runs on Update 5 without migration, code changes, or rework. Solution Center's multi-version routing allows 10.1.4 and 10.1.5 installations to coexist side-by-side during phased rollouts.

Platform Reliability

- Designer–Runtime IPC applies a bounded timeout, so a slow device driver can never drop an active Designer session.
- Backend services maintain uninterrupted operation in unattended deployments.
- Protocol driver file lookup is case-insensitive across Windows and Linux/Docker hosts, enabling seamless cross-platform deployments.

Enterprise IT, Security, and Deployment

OIDC / OAuth2 Single Sign-On
FrameworX now integrates with enterprise identity providers through OIDC/OAuth2, with out-of-the-box support for Keycloak, Auth0, Microsoft Entra ID, and login.gov. Per-provider configuration dialogs, LDAP dispatch, and industry-standard JWT validation (RFC 8725) are all managed by the runtime. The SSO sign-in button on the client logon surface is available from Update 5b onward.

Runtime REST API v1.0
A new data-plane REST API (port 3101+) exposes live tag values, alarm state, historian queries, and aggregated history — purpose-built for enterprise IT integrations and partner systems. Authentication is via Bearer-GUID. Update 5b adds tag write support.

SolutionCenter API v1.0
A management-plane REST API (port 10108) covers the full solution lifecycle: solution management, file operations, license control, and machine-level administration. Secured with JWT/OIDC and documented via a published OpenAPI 3.1 specification. *Availability note: the API surface shipped in this update, but activation is deferred to a future FrameworX release — the API cannot be enabled on 10.1.5 builds.*

Kubernetes-Native Deployment
TServer and TWebServices expose `/health` and `/ready` HTTP endpoints fully compatible with Kubernetes liveness and readiness probes, making container-native and cloud-hosted FrameworX deployments first-class citizens.

PowerShell and CLI Tooling

- **Tatsoft.FrameworX PowerShell Module v1.0.0** — wraps the installer, TManageServices CLI, and SolutionCenter API into a cohesive scripted deployment toolkit.
- **UpgradeSolution.exe** — command-line solution upgrader with automatic named backups; the original solution is fully preserved if any step encounters an issue.
- **TManageServices CLI** — script-friendly Windows Service install/uninstall with standard exit codes and silent mode.

Security

- `Session.AutoLogOff` correctly enforces the configured `DurationHours` value across all client types.
- `TPasswordBox` delivers improved input reliability: cursor positioning is preserved during mid-string edits, and a binding adapter is available for tag write-back.
- `SecuritySecrets` values are masked by default; clipboard operations are restricted to administrators. `BearerToken`, `AssociatedUser`, and `Enabled` fields are accessible programmatically.
- ADO API database operations include built-in SQL injection prevention; an opt-in parameterised query mode via the `PARAMS` prefix enables safe handling of untrusted inputs.
- TLS/SSL certificate management for remote engineering workflows is more robust; MQTT certificate upload is available in a unified Designer dialog.
- `EditSecurity` popup management is improved: the dialog responds correctly to repeated interactions and permission-denied messages accurately reflect the access state.

Historian, Alarms, and Runtime Stability

Historian — Full Nested Path Support
Historian can now be enabled for tags in any AssetTree subfolder structure with full path fidelity. When enabled via Tag Properties, the complete qualified path (e.g. `Folder1/Folder2/Tag`) is stored and used for history retrieval — ensuring that deeply nested assets are historised, queried, and displayed with complete accuracy.

> ⚠ **Upgrade note:** Tags previously enabled with a leaf-only name will appear as *"not enabled"* when Tag Properties is reopened on Update 5. Re-enabling via Tag Properties registers the correct full path. Existing historical data is not affected.

Historian and Canary Compatibility
Canary Historian maintains continuous logging through v9.2-to-v10.1 solution upgrades. The AssetTree path is preserved consistently when enabling Historian, Alarm, or Device for nested tags.

Runtime Stability

- Communication layer improvements provide significantly stronger TCP connection resilience under sustained high-frequency polling loads.
- `OnlineConfiguration` operates cleanly without TraceWindow noise.
- Device group startup is robust to edge cases in sort key configuration.
- Redundancy activates correctly when a solution is started from Solution Center; SmartClient URLs correctly carry redundancy parameters.

Alarms

- AlarmViewer displays sub-second (millisecond) timestamps with full precision when configured.
- AlarmViewer theme changes (Light/Dark) propagate correctly at runtime.
- AlarmViewer operates correctly in solutions where the Alarm module is present but not currently active.

Trend Chart and DataHub

- Quality-gap rendering is consistent between WPF and HTML5: gaps appear correctly when adjacent point quality transitions to Undefined.
- Each pen independently displays its own value for uncertain-quality (quality 64) data.
- Pen unit settings are preserved per pen after navigating away from a trend view and returning.
- Legend text respects the active theme on both WPF and HTML5.
- Trend Chart symbols render correctly in printed reports.
- `LoadXY` method behaviour is consistent between WPF and HTML5.
- DataHub cross-pen value isolation is improved for quality change events.

Devices and Protocols

OPC UA

- Tag imports complete reliably for address spaces beyond 1,000 tags, with no impact on Designer responsiveness during large import operations.
- Private UDT members are correctly withheld from external OPC UA clients, preserving data confidentiality at the OPC UA layer.
- Remote engineering certificate management correctly generates and binds certificates on the remote runtime host.
- A new *Host Override* field on the OPC UA Client enables flexible endpoint configuration for complex network topologies.

ControlLogix

- Tag imports complete reliably for large databases beyond 1,000 tags.
- Multi-byte data types are read and written with correct byte ordering in all configurations.
- Program-scope tag communication is fully supported via CIP two-segment encoding; `<Type>:PROGRAM:<TaskName>.<Member>` addressing is parsed correctly.
- A new `ForceCIPLittleEndian` option per Point enables compatibility with non-standard device configurations.
- L5K file import is now supported.

Siemens

- The XML importer produces consistent tag mappings, valid data types, and correct PLC addresses across all configurations.
- DBSTR/DBST/ST length is enforced as the total byte count including the 2-byte S7 STRING header (`N+2` for TIA `STRING[N]`), and fully documented.
- Legacy address formats are automatically migrated to the canonical form when the configuration dialog is reopened.

MQTT

- TLS 1.1 and TLS 1.2 connections are fully supported.
- The Sparkplug B `PublishMetricNames` option decouples name-based publishing from Store and Forward — third-party subscribers that join after DBIRTH receive named DDATA regardless of Store and Forward state.
- Certificate upload is available through a consolidated, unified Designer dialog.

Other Drivers

- **DNP3**: log path is configurable; TraceWindow output correctly identifies the device.
- **AB SLC/PLC5**: Byte Swap is correctly applied to Device Points over TCP/IP.
- **TwinCAT**: stale handle caches are cleared on reconnect; the driver recovers from ADS router restarts without requiring a runtime restart.
- **Omron FINS Master**: UDP interface is fully supported.
- **ModbusSlave**: startup is clean when a channel uses the ModbusSlave protocol.
- Channel-level disable correctly scopes to the intended channel without affecting sibling channels.
- Device Points scaling values propagate correctly to the mapped tag definition.

Scripts and Automation

Script Engine

- `@Module.Property` references compile correctly in scripts that combine multiline block comments (`/* ... */`) and single-line comments (`//`) on the same line.
- Script Classes automatically inject WPF usings (`Canvas`, `TDisplayControl`, and related types), eliminating boilerplate in display code-behind scripts.
- Script Tasks in solutions with namespace classes compile cleanly; cross-script references to Script Class types resolve correctly at compile time.
- Compile diagnostics surface the fully qualified name (Namespace.Class) of unresolved identifiers, making troubleshooting faster.
- Designer *Rebuild All* converges correctly across cross-row Script Class DLL dependencies.

New Script APIs

- `@Server.Retentive.Snapshot()` — forces an immediate retentive flush on demand.
- `AddTagAndDevicePoint` — programmatic provisioning of tags and device points from scripts.
- `TK.ApplyImageBytesToShape` — applies an in-memory image to a Shape fill at runtime.
- The Script TK toolkit is reorganised into six sub-namespaces for improved discoverability. All existing `TK.X` scripts continue to work without modification.

Visualization, Displays, and Reports

New Controls

- **TRadialGauge** — a single portable control (Full, SemiCircle, CenterValue, and BandedRange presets) available across WPF and HTML5. Replaces three legacy circular gauges while maintaining backward compatibility for existing solutions.
- **TChatSession** — renders AI conversation threads bound to session tags, enabling operator-facing AI chat interfaces on any display.

Display Authoring and Runtime

- Image flip operations and clockwise polyline rotation behave correctly in all display configurations.
- Symbol Width/Height edits apply immediately in the Properties panel; a new `LockAspectRatio` property provides precise aspect-ratio control.
- Dashboard displays with static-text TextBox cells compile and render correctly.
- `TChildDisplay` frames remain correctly populated when a parent display is revisited.
- Display navigation buttons respond reliably during display load.
- Visibility transitions operate correctly on components with combined Shine and Visibility dynamics.
- `TTextBlock` MaxLength is configurable directly in the Designer.
- Display Connectors are fully restored with undo, clipboard, and delete support; endpoints snap to connectable shapes and auto-finalize under the SelectionTool.
- Display open time is improved with a cold-load optimisation.

TWebBrowser

- A new `EnableBrowserStorage` property enables localStorage support.
- SVG rendering handles URL-encoded colours and missing viewBox attributes correctly.

HTML5 Web Client

- DataGrid columns persist after display navigation.
- Blazor-based elements load correctly on mobile browsers.
- SSL/TLS URLs without a trailing slash connect correctly.
- DateTimePicker calendar and clock popups display correctly within `TChildDisplay` containers.
- A `/displays/resource/<name>` HTTP endpoint enables image resolution for embedded viewers over plain HTTP.

Reports

- PDF, XPS, and HTML report save is fully supported on .NET 10 deployments via a co-shipped .NET 4.8 helper.
- Text, Unicode, and ASCII export correctly preserves table content.
- The new Alarm Linear Gauge report element provides alarm-state visualisation in printed reports.
- Trend Chart symbols render correctly in all report formats.

Mobile

- First iOS Camera integration for the Mobile FX Client: QR/barcode scanning and photo capture are accessible via the `ClientSession` script API.
- The PDF viewer renders correctly on both mobile browsers and the native mobile app.
