Update 5d - Detailed Release Notes

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

FrameworX 10.1 Update 5d — August 11, 2026

Build: 2093

AI, MCP and automation

- **Author display configuration completely from automation** — legend, connections, chart orientation, child-display links, and themes.
- **Update a display's code-behind script while leaving the drawing canvas untouched** — content-only display import.
- **Import a display page from JSON with full fidelity** — nested groups, dialog settings, and exact layout preserved.
- **Writing a display from automation updates its elements in place**, even while the display's editor is open.
- **Read several displays in a single full-detail request.**
- **Create tags directly inside UNS folders from automation** — `AddTagAndDevicePoint` accepts path-bearing tag names.
- **Verify dialog content and prompts in automated Designer tests**, without rendering them.
- **Drive and complete suppressed Designer configuration dialogs** in automated tests, without rendering them.
- **Drive and observe the running operator clients** — WPF and web — from automated tests, including over the HTTP AI Runtime connector for parallel test runs.
- **Richer automated UI gestures** — press-and-hold, hover, and true double-click.
- **Inspect large displays in context** — compact element-level tree summary, filters, and paged object reads.
- **Validate electronic-signature and security workflows with automated tests**, safely.
- **Concurrent AI sessions keep their exported solution files isolated.**
- **Importing a large ontology returns a concise, readable summary.**
- **The built-in AI chat can act through an external MCP tool server**, scoped to the signed-in user.
- **Recursive (self-referencing) template definitions are caught at build time** with a clear warning.
- **Solution exports record the full path the solution was created at**, not just its name.
- **MCP server hardening batch** — a set of robustness and safety improvements across the MCP server surface.

Local AI and AI Chat

- **Local AI endpoint configuration is easier to get right** — whitespace is trimmed and sensible defaults (URL, headers, secret name) are pre-seeded when a provider preset is selected.
- **Updated Claude model suggestions and clearer AI endpoint Test results.**
- **AI Chat context grounding uses a dedicated field**, keeping the question and reply tags cleanly separated.
- **AI Chat keeps a stable, continuous conversation** and answers with verified information, referencing only the tools it actually has.

Displays and Symbols

- **Symbol size, rotation, and legend are preserved through a JSON display import.**
- **Option to hide the border of a symbol's legend.**
- **Library symbols with a theme-color FillColor dynamic render correctly in the Designer editor** — runtime rendering was already correct.
- **Fill color is applied to the intended object** when another object is clicked immediately after.
- **Canceling a drawing tool mid-draw leaves the canvas clean** — no leftover connections, orphan shapes, or extra undo entries.
- **Selecting a button whose Action script method was renamed or deleted keeps its Action target intact.**
- **Editing a button's Action settings enables the Save button in Designer.**
- **Label control keeps its Contents Align setting across page navigation.**
- **The Shiny MouseOver effect works in WPF** when the button also opens a display.
- **TTextBox supports text wrapping.**
- **FileExplorer control** — refreshed appearance, remembers the last selected folder, and the Directory Tag navigates to the full path.
- **The Sidebar opens from the CodeBehind display editor.**
- **Display Connectors infrastructure is available.**
- **Folder-path tag bindings compile correctly in Canvas display code-behind.**
- **Trend Chart and Drilling Chart show accurate live data on dual-value (XY) pens.**
- **Trend Chart stays populated** when navigating between displays.
- **Trend Chart captures rapid value spikes and quality changes in online mode.**
- **Trend Chart renders quality transitions cleanly** — tags at Quality 64 keep their own value between two Quality 192 points.
- **The AlarmsTrendLayout template uses the correct theme-toggle icon and a theme-aware header.**

HTML5 and web client

- **The HTML5 web client loads significantly faster** with compressed delivery and browser caching.
- **Web clients connect smoothly through reverse proxies and Cloudflare Tunnel.**
- **Web Display scripts run** even when a referenced Script Class had not yet been built for the web.
- **TDataGrid columns populate** regardless of how FieldName casing matches the DataTable column name.
- **HTML5 BarGraph objects reflect the live value**, filling from zero and clearing when the value returns to zero.
- **Bar Graph honors the configured Fill (%) range in HTML5.**
- **Trend Chart labels stay readable on dark backgrounds in HTML5.**

- **Display CodeBehind's `Clipboard.SetText` guidance points to a method that compiles.**

Scripts and Code Editor

- **Multi-line string literals and string interpolation with Tag/Info references compile correctly** in Scripts and Display CodeBehind.
- **Server-domain Script Classes build correctly** when a method contains a nested interpolated string.
- **Calling a private or protected Script Class method reports the precise CS0122 diagnostic.**
- **IntelliSense exposes every class inside a Script Namespace** and requires instantiation for non-static methods.
- **IntelliSense refreshes and recognizes newly created Classes and Namespaces.**
- **IntelliSense and the Dynamics method list keep working** with tuple destructuring in a `foreach` header.
- **Creating a namespace in Scripts/Classes keeps `using` directives clean.**
- **Third-party DLLs in Script References load correctly** alongside a same-name bundled library.
- **Find All References is available in the Script Class Code Editor.**
- **The Code Editor keeps cursor and scroll position** when switching tabs or saving a script.
- **Comment or uncomment selected lines with a keyboard shortcut.**
- **Deleting a class or open document closes its editor tab.**
- **Script Editor matches Display CodeBehind editor behavior** for saved-line highlighting and `#region` state.
- **Client Code Behind recognizes scripts and classes from a Server Namespace.**
- **Write rows back to a configuration table from a Server script.**
- **Add Trend pens for numeric tags** — reading a tag's Min/Max works from scripts.

Unified Namespace, Tags and Ontology

- **Importing an ontology (RDF/OWL) creates the object relationships and the asset-tree hierarchy.**
- **The ISA-88 ontology import template's mapping rules are applied.**
- **Exporting and re-importing a UNS graph preserves the asset hierarchy** — including folder nodes — when entities carry a source IRI.
- **Importing tags and displays together resolves display bindings to the new tags.**
- **The Asset Tree lists folders and tags in order** after importing an existing solution's UNS.
- **Initial values for GUID tags are retained** when set through the Asset Tree view or at tag creation.
- **Tag rename in the asset tree works when only letter case changes.**
- **EngWrapper API — set a tag's per-instance property overrides** (StartValue, Min, Max, Disable, DisplayText, Description, Enumeration) on UserType member instances.
- **Ontology import policy files ship with the installed product.**

Devices and protocols

- **New Toyopuc (JTEKT) PC10G Ethernet driver** (Computer Link).
- **New SIC Marking laser communication driver.**
- **PEER Protocol Driver** — communication driver.
- **Store-and-forward (local buffering) for the Azure IoT Hub / MQTT driver.**
- **DataHub Tag Provider brings tags from devices running outside the TServer online.**
- **OPC UA: the UDT array member `Value` is sent by the OPC Server.**
- **Modbus TCP master sends well-formed frames and reads devices reliably**, and the Modbus master flags a short response instead of filling missing values with zero.
- **Siemens "TIA Portal (local)" tag import opens reliably** and covers PLC tag tables (Input/Output/Memory) as well as Data Blocks.
- **Siemens Plus tag import shows .xml files in the browse filter.**
- **Siemens Plus writes STRING (DBSTR) values within their exact bounds**, keeping the adjacent PLC address intact.
- **MQTT Client Connector keeps CPU steady on invalid broker credentials** and stops cleanly after Stop Solution.
- **Device points configured for a channel are available to the runtime.**
- **The `@Device.Channel.<name>.Start` expression resolves unambiguously** in Designer IntelliSense and at runtime.
- **CODESYS PLC Handler preserves the protocol and gateway protocol settings** in the station editor.
- **The Server-to-Server connection brings historical data across** through the Tag Historian option.

Alarms, Historian and redundancy

- **Historian and Alarm redundancy replication keeps up on large projects** — the standby database synchronizes fully.
- **Historian and Alarm data is drained to the standby before a redundancy handover completes.**
- **The Historian module starts reliably when logging to large Normalized tables** on TimescaleDB / PostgreSQL.
- **Alarm history keeps recording** through a configuration change and while the alarm-history database is briefly unreachable.
- **Alarm Viewer respects the custom alarm-history table name**, and single alarms save correctly on SQL Server historians.
- **Diagnostic tools can connect to the standby server** in redundant systems.
- **The standby server keeps the Devices module aligned with a device's Initial State** (Reserved or Remote).
- **Documented the supported redundancy configuration for Docker bridge-mode deployments.**

Runtime and platform

- **The .NET 10 runtime starts with all modules in place.**
- **The FrameworX runtime runs continuously on macOS** with the standard launcher.
- **Client tag preload and first access are faster** through the TK API (WPF and HTML5).

Security and access

- **OIDC single sign-on (HTML5)** signs users in through a correctly configured OIDC identity provider.

- **PostgreSQL dataset connections can enforce per-operator row-level security**, opted in per database.
- **Security secrets written through the AI tools are stored encrypted**, with password fields protected on every tool surface.
- **Run Security set on a display from automation is stored.**
- **Security identity providers created from automation store the chosen authentication type** (LDAP or OIDC).
- **Permission values stay intact** when a permission group name is unknown or empty in table imports and grid editing.
- **Machine settings file reads and writes are scoped to the settings folder.**
- **AI knowledge asset downloads validate server certificates** and are bounded in size and redirects.
- **Import/Export hardening batch.**

Designer, Solution Center and packaging

- **Designer opens Solution Capabilities immediately** the first time.
- **Solution Center licensing calls are more resilient** — connection reuse, recovery after network loss, and forward-slash path handling.
- **MCP configuration templates are consistent**, with a clear label in Execution Profiles.
- **Git Import restores projects with a clean build.**
- **Changing the Target Platform behaves correctly** in the engineering environment.
- **The Data debugger inspects a DataRow like a DataTable**, and large DataTable views stay within the screen.
- **Upgrading FactoryStudio 9.2 projects to FrameworX 10.1 preserves UserType (UDT) definitions** and imports User Types correctly.

Breaking Changes and Deprecations

The SolutionCenter API can no longer be enabled on a released build. The API has shipped disabled by default since 10.1.5 GA, pending security hardening work on the underlying service surface. Until this update, an operator could override that gate by setting both `SolutionCenterApi.Enabled` and `SolutionCenterApi.AcceptRiskOnOldBuild` to `true` in `MachineSettings/TWebServices.json`. That override is no longer available in a released build, and no configuration setting re-enables the API in the 10.1.5 line. Installations that had enabled the API this way will find it stops responding after this update. The API remains planned to become available in a future release once the hardening work completes.

Windows-Authentication logins relayed through a Secure Gateway now require the gateway's Windows-Authentication mode. The runtime honors a Windows-Authentication claim as proof of identity only when the connection itself was Windows-authenticated. If you run `TSecureGatewayService` with `WA=false`, set `WA=true` so the gateway relays over the runtime's Windows-Authentication port — otherwise Windows-authenticated users through that gateway are refused after this update. Password-authenticated users through the same gateway are unaffected. The runtime must be started with `/wa` on .NET Framework for the Windows-Authentication port to exist.

Lifecycle and Support

Update 5d ships on **August 11, 2026** as a **free upgrade for every FrameworX 10.x customer**, with the platform's standard within-major-version 100% compatibility commitment. See **Breaking Changes and Deprecations** above for two configuration-level changes that may require an adjustment on affected installations.

Security note

This release contains security-relevant corrections. Upgrading is recommended for all installations.

Two items require action beyond upgrading:

- If you configured display security through automation on 10.1.5c or earlier, review those displays and re-apply Run Security. Upgrading does not re-apply it retroactively.
- Rotate any credentials that were written through the AI tools on 10.1.5c or earlier. Upgrading protects new writes; credentials already stored remain exposed until rotated.

Customers who need per-item detail — affected versions, severity, and required action — can request it from **support@tatsoft.com**, along with the component manifest (CycloneDX) for the version they are running.
