Update 5d

This was the current update for FrameworX version 10.1 prior to the 10.1.5e release. Published on August 11, 2026. Version 10.1.5d (build 2093). It follows Update 5c (June 22, 2026) and concentrates on AI-driven authoring and automated testing, display and symbol fidelity, the Scripts and Code Editor experience, ontology round-tripping, redundancy reliability on large projects, and faster web delivery.

FrameworX 10.1 Update 5d — August 11, 2026

The Update themes:

- **AI authoring and automated testing** — author complete display configuration from automation, update a display's code-behind without touching its canvas, full-fidelity JSON display import, and drive suppressed Designer dialogs, electronic-signature workflows, and running WPF and web operator clients from automated tests.
- **Display and symbol fidelity** — symbol size, rotation, and legend survive a JSON display import; theme-color symbol rendering is correct in the Designer editor; drawing-tool cancel leaves the canvas clean; and Trend Chart shows accurate live data, including rapid spikes and quality transitions.
- **Scripts and Code Editor** — multi-line string literals and interpolation with Tag/Info references compile correctly, IntelliSense recognizes newly created classes and namespaces, Find All References is available in the Script Class editor, and the editor keeps cursor and scroll position across tab switches and saves.
- **Unified Namespace and ontology** — RDF/OWL imports create object relationships and the asset-tree hierarchy, the ISA-88 import template's mapping rules are applied, and a UNS graph round-trips through export and re-import with its hierarchy intact.
- **Devices and protocols** — new Toyopuc (JTEKT), SIC Marking laser, and PEER drivers; store-and-forward buffering for the Azure IoT Hub / MQTT driver; Siemens tag import extended to PLC tag tables; and OPC UA UDT array members sent by the OPC Server.
- **Alarms, Historian and redundancy** — redundancy replication keeps up on large projects with the standby fully synchronized, alarm history keeps recording through configuration changes, and diagnostic tools can connect to the standby server.
- **Web client** — significantly faster HTML5 load through compressed delivery and browser caching.

See Update 5d - Detailed Release Notes for the complete item-by-item list.

AI, MCP and automation

This release makes AI-driven authoring safe to use on existing solutions and extends automated testing across the whole product.

- **Display authoring from automation is complete** — legend, connections, chart orientation, child-display links, and themes can all be configured programmatically, and a display's code-behind can be updated without altering the drawing canvas.
- **Full-fidelity JSON display import** — nested groups, dialog settings, and exact layout are preserved.
- **Automated testing reaches the whole product** — suppressed Designer configuration dialogs, dialog content and prompts, electronic-signature and security workflows, and the running WPF and web operator clients can all be driven and observed from automated tests, with richer gestures and in-context inspection of large displays.
- **Concurrent AI sessions are isolated** — exported solution files stay separate between parallel sessions.
- **Security secrets written through the AI tools are stored encrypted**, with password fields protected on every tool surface.

Displays and Symbols

Symbol size, rotation, and legend now survive the JSON display import round trip. Library symbols with a theme-color FillColor dynamic render correctly in the Designer editor, fill color is applied to the intended object when another object is clicked immediately after, and selecting a button whose Action method was renamed keeps its Action target intact. Trend Chart shows accurate live data on dual-value (XY) pens, stays populated across display navigation, and captures rapid value spikes and quality transitions in online mode.

Scripts and Code Editor

A broad pass over the scripting experience: multi-line string literals and string interpolation with Tag/Info references compile correctly in both Scripts and Display CodeBehind; IntelliSense exposes every class in a Script Namespace, refreshes for newly created classes, and works with tuple destructuring in a `foreach` header; namespace creation keeps `using` directives clean; and third-party DLLs in Script References load correctly alongside a same-name bundled library.

Unified Namespace, Tags and Ontology

Ontology import and export round-trip correctly: RDF/OWL imports create object relationships and the asset-tree hierarchy, the ISA-88 template's mapping rules are applied, and an exported UNS graph preserves its asset hierarchy on re-import when entities carry a source IRI. In the asset tree, GUID tag initial values are retained, case-only tag renames work, and folders and tags list in order after importing an existing solution's UNS.

Devices and protocols

Adds the **Toyopuc (JTEKT) PC10G**, **SIC Marking laser**, and **PEER** communication drivers, plus store-and-forward local buffering for the Azure IoT Hub / MQTT driver. Siemens tag import now covers PLC tag tables (Input/Output/Memory) as well as Data Blocks and writes STRING values within their exact bounds. OPC UA UDT array members are sent by the OPC Server, Modbus TCP master frames are well-formed, the DataHub Tag Provider brings online tags from devices running outside the TServer, and the MQTT Client Connector keeps CPU steady on invalid broker credentials.

Alarms, Historian and redundancy

Historian and Alarm redundancy replication keeps up on large projects, so the standby database synchronizes fully, and 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 configuration changes and brief database interruptions, diagnostic tools can connect to the standby server, and Alarm Viewer respects a custom alarm-history table name.

Web client and performance

The HTML5 web client loads significantly faster through compressed delivery and browser caching, connects smoothly through reverse proxies and Cloudflare Tunnel, and client tag preload and first access are faster through the TK API. The runtime starts with all modules in place on the .NET 10 build and runs continuously on macOS with the standard launcher.

Security and access

OIDC single sign-on signs HTML5 users in through a correctly configured identity provider, PostgreSQL dataset connections can enforce per-operator row-level security, and machine settings file access is scoped to its intended folder. Security identity providers and secrets authored through the AI tools store their type and stay encrypted.

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 — every 10.1.x solution opens and runs without migration or rework. See **Breaking Changes and Deprecations** above for two configuration-level changes that may require an adjustment on affected installations.

The full lifecycle policy, version support phases, and end-of-support dates for older FrameworX and FactoryStudio versions are documented on the Product Lifecycle and End of Support page.

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.

- Update 5d - Detailed Release Notes