Update 6a - Detailed Release Notes

Complete list of features, improvements and corrections in FrameworX 10.1 Update 6a (version 10.1.6a), by product area.

Update 6a installs on top of 10.1.6, in the same installation folder. Solutions keep running without migration or rework, and every installation on 10.1.6 can move to Update 6a.

FrameworX 10.1 Update 6a, October 28, 2026

Version: 10.1.6a. Summary and behavior changes to review: Update 6a.

Unified Namespace and knowledge graph

- **Solutions query graph databases with SPARQL.** A REST WebData object holds the endpoint and the query of any SPARQL 1.1 graph database, and results arrive as a DataTable ready for a display. A multi-hop question about the plant becomes one query.
- **Imported ontologies carry their relationships, dates and member values into the solution.** The AI tools, the RDF export and the UNS visual show them, and the import lists every element it does not map.
- **Dates written to DateTime tags and UserType defaults keep their time zone.** Values written through the AI tools, an ontology import or a JSON import are stored in ISO 8601 and load the same on any regional setting.

Devices and connectors

- **The Value Simulator produces realistic process data.** Each point generates sine, cosine, square, triangle or ramp waves with its own period, values that follow a setpoint like a controlled process, value lists, decimal ranges and a chosen quality. Demo and training solutions show plausible values with no device attached.
- **The BACnet connector reads Schedule objects and individual status flags, and writes at a chosen priority.** A channel option sets the priority of Present Value writes, with relinquish, for building-automation integrations.
- **The Altus ALNET II connector runs on Windows and Linux, ARM64 included, with no external library.** Unsolicited writes from the controller update exactly the addressed points.
- **The SNMP connector runs on Linux and in Linux containers, ARM64 included.** MIB files selected in the Designer on Windows are found when the solution runs on Linux.
- **Modbus Slave channels answer continuously through online changes, ResetModule restarts and redundancy switches.** A Modbus Master keeps its unit ID when the slave channel it polls restarts.
- **The OPC UA Server publishes only the asset folders you choose.** Set an asset folder to Private and the server hides the folder and everything under it, DataHub and Tag Provider tags included.
- **The OPC UA TagProvider reconnects with one session on the server.** It closes its previous session when it reconnects.
- **The Modbus and Control Expert simulators start cleanly from Devices > Channels.**
- **The Kafka connectors ship an updated libcurl.**

Web client (HTML5)

- **Displays and symbols open faster.** The display loader resolves each element type once per display, and displays with content written by hand or pasted from other WPF tools load reliably.
- **The Alarm Viewer and the DataGrid stay responsive on large tables.** An editable grid of 2000 rows opens in about 3 seconds, where it took about 20. Blinking alarms blink as on the desktop, and a grid-settings tag keeps the operator's sort and filter as data updates.
- **Data grids edit CheckBox and ComboBox columns, as in the desktop client.** ComboBox columns use a drop-down list built from the column's Format, and cells you did not change keep their stored values.
- **Display code reads tag values on first open.** The web client receives the tag values before it creates the display's code, so lists filled from a tag show the right items from the start.
- **Display actions start and stop device channels, as in the desktop client.** Device.Channel.<Name>.Start and Stop work from a button, and the channel's IsRunning property shows when it runs.
- **The Trend Chart cursor shows its line and values box with alarm markers on the chart.** Cursors keep working while you pan.
- **A click anywhere on a ComboBox's line opens its list, with or without a label.**
- **On Windows, .NET 10 solutions show web client grids and charts in the solution's language.** Texts, dates and numbers follow the solution's culture, and UNS Visual reports generated by a server script show their graph.

AI, MCP and automation

- **The AI tools report the exact result of every action.** This covers dialogs, a Designer close, alarm and history answers, drop-down edits, a runtime start, right-click menus and buttons that act on the mouse press. A write names any value it did not apply, with the reason.
- **Changes to a running solution on Production, Validation or Custom profiles stay under your control.** An AI call asks you to stop, start or hot-reload the runtime, change configuration online or step the script debugger when the profile is not Development. Your own clicks work as always.
- **The AI Designer keeps the execution profile you selected.** A runtime you started on Production, Validation or Custom stays connected while an AI session works, and the Designer stays responsive.
- **Shorter and accurate tool descriptions leave more of the AI model's context for your work.** Smaller local models can now use the tools.
- **The AI Designer works with Windows message boxes, file dialogs and print dialogs.** It sees them in its state and screenshots and answers them by button name, including a box that opens while one of its own actions runs.

- **Dialogs from your own clicks and work come to you while the AI Designer works.** The AI action waits until you close them.
- **The AI Designer exports, installs, updates and removes plugins.** It saves a plugin as a .dbplugin file and installs it as a merge or read-only, with or without a prefix. A read-only plugin stays installed while the solution uses one of its objects.
- **The AI Designer drives the device configuration grids.** It selects several rows at once, opens cell menus, runs Edit Combined Rows and fills the Edit Multiple Rows dialog. Screenshots include open popups.
- **Opening a solution through the AI Designer answers within the AI client's wait.** A slow-starting Designer reports that it is loading, a maximized Designer stays maximized, and the right Designer comes to the front.
- **AI grid and page actions change only the objects the AI is allowed to modify.**
- **The AI tools show tags in folders by their full name in instance lists and search.**
- **AI Chat names the equipment a link member points to.** External AI tools connected through the AI Runtime receive the same answer.
- **The AI Designer matches each display mode to its engine.** A PopupWindow display written for the Portable or HTML5 engine is saved as Popup, and the reply says so.
- **Graph export and the UNS visual follow "Enable MCP for Designer" and read-only solutions.**
- **The AI Runtime switchover tool is available on request.** The person who sets up the AI Runtime server turns it on with an explicit option, and AI clients confirm before it runs.
- **The AI Runtime client tools respect electronic signature and work with clients on the same machine.**
- **AI assistants see the tools built for solution work.** Tools used for Tatsoft's own automated testing are kept out of their list.

Displays and Symbols

- **Editable ComboBox accepts typed values and keeps its value when you return to a display.** A value outside the list goes to the Linked Value tag, and the desktop client keeps typed text in the case it was typed.
- **The AssetsTree control filters its tree by UserType.** Link the UserType Filter setting to a tag that lists type names, and operators see only the tags of those types and the folders leading to them.
- **Scripts turn the on-screen keyboard on or off with @Client.OnScreenKeyboard.** On the Rich Client and Smart Client the change applies at once to every display.
- **The Draw editor opens displays at a default zoom you choose.** The zoom is a per-user Designer setting, from 10% to 800%, for small screens and presentations.
- **Display parameters with more than one part, such as V=10 / 4, show the computed value.**
- **Code on the startup display's opening event runs when the client starts.**
- **Rich Client display code-behind reads tag values on every start.**
- **The Rich Client stays fast over long sessions.** Alarm Viewers stop with the displays that hold them, so navigation keeps its speed after redundancy switches.
- **Copying symbols with images inside a solution reuses the original symbol.** Pasted instances keep their link to it.

Operator input and electronic signature

- **A Slider asks for the electronic signature once, when the operator ends the action.** It writes the final value and adds one audit trail entry.
- **Move/Drag objects follow the same security rules as other operator inputs.** A confirmation or signature is asked once, when the drag ends.

Expressions and scripts

- **Expressions evaluated at run time follow VB.NET rules.** Display and popup parameters, trend pen expressions, alarm messages, report file names, dataset texts and TK.EvaluateExpression combine numbers, texts and Booleans without explicit conversions, and accept VB.NET operators and functions such as &, Mod, Like, IIf and Format. Values read with Asset(...) work directly in arithmetic.
- **Allow Client Tasks controls client tasks in WPF clients.** ClientStartup, ClientShutdown and Client-domain tasks run when the option is checked.
- **The code editor remembers which #region blocks you left open or closed in each script.** A link to a method inside a closed block, such as a button's Action > Edit, opens the block.
- **A Reference member with target type Dynamic links to tags of user-defined types.**
- **@Client.Session.SetBlockedUserAsync compiles without a warning.**

Designer

- **Several Designers and the runtime share a solution with every edit kept.** The solution server reloads when it cannot confirm its cache, and tag values written from the Designer carry the current timestamp.
- **Solutions built from JSON files keep display bindings to Dataset Tables.** To refresh a solution built earlier, import its files again.
- **The automatic Git export writes the whole table when you leave a page.** Exporting tags to JSON or ZIP is faster on solutions with many asset folders.
- **Bulk import creates missing UNS parent folders and places every asset row.**
- **Importing a plugin with a prefix brings in the plugin's Startup layout.** Mobile pages of imported layouts point to the imported displays.
- **Enter saves the password in the password editor.**
- **Display thumbnails built with Validate Displays show the password box label.**
- **The Brush Editor image picker shows the selected image's name and scrolls to it.**
- **The GridLine object shows its own name.**
- **The DataHub tile on UNS > Data Servers is titled "Publish to a DataHub Service" and shows its endpoint.**
- **Solution Center and Designer messages show in the selected language.** Korean, Chinese and Turkish dialogs show their own text.

Runtime, redundancy and performance

- **Solutions with many tags use less memory, and tag changes are processed faster.** Value conversions give the same result on .NET Framework 4.8 and .NET 10.
- **The runtime restarts on Server.Restart and recovers automatically after an unexpected stop.**
- **A redundant standby keeps its startup options through solution updates.** It logs each update and keeps one Rich Client open.

- **The Trace Window and other Windows clients reconnect on their own after a server stall.** Closing the Trace Window ends its process.
- **The communication layer handles every method shape and copies less data per call.** Closed and timed-out connections are handled cleanly, which keeps the logs clear.
- **The Historian on SQL Server stores every sample, including several within the same second.** MongoDB queries handle decimal values, and online changes to user types keep every row.
- **The Solution Manager service counts remote license seats per runtime.** It handles concurrent file transfers and releases gateway Tag Provider connections on reconnect.
- **Runtimes started inside a container by Solution Center or the Designer run in Docker mode.**
- **Installing the runtime Windows service from the command line states the reason when the install does not complete.**
- **The Runtime REST API describes a tag in a folder by its full path.**
- **Property Watch edits array tags element by element.** Arrays with three or more dimensions are listed one element per row.

Reports

- **Report content shows expressions that use &, < or >.** Tag values inserted into report content are shown as text.
- **The Report Viewer shows saved reports with the values they were saved with.**

Installation

- **The installer finds an existing installation from the Windows record of installed programs.** A silent first installation with /DIR installs into the folder it names.
- **The expression library ships as a separate file, with its license notice beside it.**

Compatibility and Lifecycle

- Update 6a installs on top of 10.1.6. Upgrade steps: Managing Updates and Multiple Versions Reference and Install and Activate.
- Earlier updates: Release Notes. Support phases and end-of-support dates: Product Lifecycle and End of Support.

Security Note

This release contains security corrections and hardening. For per-item detail (affected versions, severity, required action) and the component manifest (CycloneDX) of your version, contact **support@tatsoft.com**.
