KnowledgeGraph Control Reference

Portable Display control that renders the solution's industrial ontology as an interactive graph.

Reference **Controls** **Charts** KnowledgeGraph

Version 10.1.5+

Overview

The **Knowledge Graph** control (XAML element `TKnowledgeGraph`) renders the UNS UserType hierarchy and per-Tag relationships as an interactive node-and-edge graph. Operators pan, zoom, and click nodes to inspect the underlying assets; node selections flow back into Tag-bindable properties so the rest of the Display can react to graph interaction.

The control reads its source from the `SolutionSettings.KnowledgeGraphSource` column — a Mermaid Markdown document regenerated by:

- The **Knowledge Graph** button on the Asset Tree (manual).
- The `generate_uns_visual` MCP tool (AI-driven authoring).
- The `TK.GenerateUnsVisual` script API (programmatic).
- The auto-refresh hook on UnsUserTypes / UnsTags / Asset Tree changes.

Rendering is performed by a vendored cytoscape.js inside a WebView2 (WPF) or OpenSilver (HTML5) host — no external network dependency at runtime.

Requirements

This component is **Portable**: it runs both on Windows WPF displays and on Web Pages on any platform. WPF deployments use WebView2 (Microsoft Edge runtime, bundled with FrameworX). HTML5 deployments render through the OpenSilver-hosted cytoscape.js, served from the Display contents.

Configuration

1. Go to **Displays / Draw**.
2. On the Components Panel, select **Charts**, then **Knowledge Graph**.
3. Click or drag-and-drop it onto the Drawing area.
4. Double-click the object to open the configuration window, or set properties directly in the Properties panel.

Knowledge Graph Settings	
Field	**Description**
Control name	Identifies the control within the Display. Used by binding expressions to address its properties.
Render mode	Selects the rendering pipeline: • **Interactive** (default) — full pan / zoom / click cytoscape.js layout. Recommended for operator workstations. • **Static** — reserved for 10.1.6. Will render a MermaidSVG snapshot for non-interactive contexts (reports, thumbnails, low-end clients). In 10.1.5 selecting this mode falls back to Interactive.
Selected node path (output)	Tag-bindable string that receives the full UNS path of the most recently clicked node (e.g. `Site1.Area2.Tank305`). Empty when no node is selected. Bind to a Tag to react to graph navigation elsewhere in the Display.
Selected node type (output)	Tag-bindable string that receives the UserType name of the most recently clicked node (e.g. `Tank`, `Pump`). Empty when no node is selected. Pairs with the path output for type-aware drill-downs.
Localize	When checked (default), node labels render in the active Designer / Runtime culture using the standard FrameworX translation pipeline. Uncheck to show raw ontology IRIs / labels regardless of culture.

Runtime execution

At runtime the control subscribes to the `SolutionSettings.KnowledgeGraphSource` column. When the column changes — the operator clicks the Asset Tree **Knowledge Graph** button, an AI session calls `generate_uns_visual`, or a Script calls `TK.GenerateUnsVisual` — the control re-renders without a Display reload. Operators can pan and zoom freely; clicking a node updates the two selection-output properties, which downstream bindings can wire to navigation actions, ChildDisplay swaps, or recipe dialogs.

Source content is the W3C-aligned Mermaid Markdown produced by the UNS Visual Report Generator (the same pipeline that emits standalone HTML / SVG reports from the Asset Tree). The control accepts `classDiagram` nodes, inheritance edges, composition edges, and the FrameworX-specific node-kind decorations the generator emits.

Source regenerators

Source	How it writes `KnowledgeGraphSource`
Asset Tree button	Operator clicks **Knowledge Graph** at the top of the Asset Tree. The visual report is generated and the column updated. Existing Knowledge Graph controls on open Displays re-render automatically.
`generate_uns_visual` MCP tool	An AI assistant invokes the DesignerMCP tool. The tool produces the same Mermaid Markdown the Asset Tree button does, writes it to the column, and returns the source so the AI can inspect or transform it.
`TK.GenerateUnsVisual` script API	A Script (Task, Class, or Display CodeBehind) calls `TK.GenerateUnsVisual()` to refresh the source under programmatic triggers (cron, alarm reaction, batch end).
Auto-refresh hook	Edits to `UnsUserTypes`, `UnsTags`, or the Asset Tree silently mark the source as stale; the next Asset Tree render or `TK.GenerateUnsVisual` call regenerates from current state.

Display action wiring

Both output properties are standard FrameworX bindable strings — the same pattern used by other portable controls' selection outputs. Typical wiring:

- Bind **Selected node path** to a Tag, then bind that Tag to the source of a ChildDisplay or a Trend Chart to drill down on click.
- Bind **Selected node type** to a Tag, then drive a TabControl or a conditional Visibility to show type-specific recipe / setpoint panels.
- Combine both in a Script Task to log the most recent operator drill-down for audit trail.

HTML5 / OpenSilver parity

The control compiles under both WPF and OpenSilver. On HTML5 the WebView2 host is replaced with the OpenSilver-hosted cytoscape.js loaded from Display contents; the property surface, source contract, and selection outputs are identical. **Localize** uses the same translation pipeline as TTextBlock in HTML5 mode.

Notes

- **Naming history.** The control was named `TOntologyChart` during the 10.1.5 preview period. The shipped name is `TKnowledgeGraph`; the old name was retired with no back-compatibility alias. Preview-period solutions persisting the old element type must re-add the control in the 10.1.5 Designer (see the Release Notes → Update 5 (Preview) Known Issues table).
- **One source, many controls.** Multiple Knowledge Graph controls on the same Display share the same source column — they all render the same graph. To show different views of the same ontology, post-process the source in a Script Task before placing it in the column.
- **Companion features.** Pairs with the Industrial Ontology Integration How-to import workflow and the LocalAI KnowledgeGraph Demo solution.
- **Design-time preview is recognition-only.** When the control is placed on a Display in the Designer, it renders a fixed 6-node hand-positioned sample (Asset Plant1/Plant2 PumpA/PumpB/TankX) rather than the live `KnowledgeGraphSource`. This is intentional — at design time, the goal is recognition ("the Knowledge Graph slot lives here"), not precision. The runtime cytoscape render is the source of truth for the customer's actual ontology.
- **Auto-fit on resize.** The runtime cytoscape graph automatically refits its viewport when the host container resizes — window resize, panel collapse, splitter drag, or any intra-window layout shift. Debounced at 150 ms. The toolbar's Fit button remains as the manual recovery path.

In this section...