Shapes Connections Reference

The **Shapes Connections** system allows you to draw connector lines between shapes and symbols in a Canvas display. Connectors are polylines that automatically snap to terminal points on shapes, creating a visual and logical link between elements. The connection data is stored as part of the display and is available at runtime for scripts, power flow calculations, and other logic that needs to traverse the connection graph.

Connections are bidirectional — both the connector line and the connected shapes store references to each other using GUIDs, so you can navigate the graph in any direction.

Overview

A connection links two elements through a connector polyline. Each connection stores:

Field	Description
Guid	The unique identifier of the connected element
Terminal	The terminal index on the connected element (which connection point was used)
Point	The coordinate of the connection point relative to the connected element
LineIndex	Indicates the direction: 1 = inbound (start of connector), 2 = outbound (end of connector)

Connections are stored as a semicolon-delimited string in the `EleDP.Connections` attached property on each element. The system manages this data automatically when you create, edit, move, or delete connectors.

Creating Connections

Using the Connector Tool

The **Connector Tool** is available in the Drawing toolbar, next to the Spline tool. To create a connection:

1. Click the **Connector** button in the Drawing toolbar.
2. Click on the first shape or symbol terminal — the connector start point snaps to the nearest terminal.
3. Move the mouse to the target shape — the endpoint follows the cursor with grid snapping.
4. Release the mouse on the target shape terminal — the endpoint snaps and the connection is created.

If the connector is too short (less than 32 pixels), it is automatically removed. Right-click to cancel and return to the Selection tool.

Using the Polyline Tool

You can also create connections with the standard **Polyline** tool. Draw a polyline between two connectable shapes. The polyline tool highlights connectable terminals as you draw. After drawing, the system creates connection records on both the connector and the connected shapes.

Terminal Points

Terminal points define where connectors can attach to a shape or symbol. They are configured inside symbols using child elements with a specific naming convention.

Defining Terminals on Symbols

Inside a symbol, add small child elements (typically rectangles or ellipses) and set their `Uid` property to `Node` followed by a number:

Uid	Description
`Node1`	First terminal (typically the input or top connection point)
`Node2`	Second terminal (typically the output or bottom connection point)
`Node3`, `Node4`, ...	Additional terminals as needed

The connector system searches recursively through the symbol's child canvas for elements whose `Uid` starts with "Node". The center of each terminal element is used as the snap point.

AllowConnections Property

For a shape or symbol to accept connections, its `AllowConnections` property must be `true`. This property is available on:

- **TSymbol** — set in the symbol's properties
- **TGroup** — set on the group container
- **TShapeGroup** — set on the shape group
- **Shape** — automatically true if the shape already has connections

---

Connector Properties

Connector polylines have the following attached properties (defined in `EleDP`):

Property	Type	Description
`EleDP.Guid`	Guid	Unique identifier for this element. Assigned automatically when the connector is created.
`EleDP.IsConnector`	bool	Marks this polyline as a connector. Set to `true` automatically by the Connector Tool.
`EleDP.Connections`	string	Serialized connection data. Semicolon-delimited list of ConnectionInfo records.
`EleDP.ParentGuid`	Guid	Reference to a parent element (used for grouped connections).
`EleDP.LegendGuid`	Guid	Reference to a label/legend element associated with this connector or shape.

The **IsConnector** checkbox is visible in the Properties panel when a polyline is selected. You can manually set any polyline as a connector.

Editing Connections

Moving Connected Shapes

When you move a shape that has connectors attached, the connector endpoints move with the shape automatically. The system captures all connections before the move and updates the connector polyline coordinates by the same delta.

Editing Connector Points

Use the **Direct Selection** tool to edit connector polylines. When a connector is selected in Direct Selection mode:

- Draggable thumbs appear at each point of the polyline.
- Drag the **first or last** point to reconnect to a different terminal — the endpoint snaps to the nearest terminal.
- **Double-click** on any point to add an intermediate bend point.
- **Right-click** on an intermediate point to remove it.
- The first and last points (connection endpoints) cannot be removed.

Connected endpoints display visual highlights: **red** when connected to a shape, **yellow** when connected to another connector.

Deleting Connections

When you delete a connector polyline, the connection references are automatically removed from both connected shapes. When you delete a shape that has connectors attached, all connection references pointing to that shape are cleaned up from the connectors.

Copy and Paste

When you copy shapes that have connections, the connected elements (connectors, legends, parent references) are included automatically. On paste, all elements receive new unique GUIDs and the internal connection references are remapped to the new identifiers, preserving the connection topology.

Undo and Redo

All connector editing operations support undo and redo. After an undo, connection endpoints are re-validated to ensure they still reference valid terminal points.

Runtime Access

At runtime, connection data is available through the display configuration. Scripts and modules can traverse the connection graph by reading the `Connections` property on display elements. Each connection record provides the GUID of the connected element, allowing navigation from any element to its connected peers.

Tip

Connections are available only in **Canvas** displays, not in Dashboard layouts. Shapes and connectors can be used inside Symbols as well.

Connection Data Format

For advanced users and script authors, the connection data stored in the `EleDP.Connections` property uses this serialization format:

text

Each entry is separated by a semicolon. Within each entry, fields are comma-separated, and the Point coordinates use `x` as the separator between X and Y values. All numeric values use invariant culture formatting.

Connector Symbols

The Connector Tool first tries to load a symbol named **"Connector"** from the project's symbol library. If a Connector symbol is found, its polyline child is extracted and used as the connector line, inheriting the symbol's stroke style, color, and other visual properties.

If no Connector symbol is found in the project, the tool creates a default gray polyline with 2px stroke thickness.

To customize the default connector appearance, create a symbol named "Connector" in your project's symbol library containing a single polyline with the desired visual properties and `IsConnector` set to `true`.

Upgrade from Version 9.2

Projects created in version 9.2 with connections are automatically updated when upgraded to version 10. The upgrade system maps the old property names to the new ones:

Version 9.2	Version 10
`DisplayComponent.GUID`	`EleDP.Guid`
`DisplayComponent.IsConnector`	`EleDP.IsConnector`
`DisplayComponent.ConnectionGUIDs`	`EleDP.Connections`
`DisplayComponent.RefToControlGUID`	`EleDP.ParentGuid`
`DisplayComponent.RefToNameGUID`	`EleDP.LegendGuid`

No manual steps are required — the conversion is automatic on project open.