PSD3 AST Grammar

This document describes the grammar of the PSD3 declarative visualization DSL, as implemented in the TreeBuilder3 interactive showcase.

Overview

The AST represents a tree of visualization specifications. Each node describes WHAT to render, and interpreters (D3, Mermaid, English) decide HOW to render it.

```
Tree datum =
  | Node ElementType (Array Attr) (Array Behavior) (Array (Tree datum))
  | Join name key data template:(datum -> Tree datum)
  | NestedJoin name key data decompose template
  | SceneJoin name key data template enterBehavior updateBehavior
exitBehavior
  | SceneNestedJoin name key data decompose template enterBehavior
updateBehavior exitBehavior
```

Top-Level AST Nodes

Key	AST Node	Color	Description
e	Node (Elem)	Gray (#6B7280)	Element with attrs, behaviors, children
j	Join	Yellow (#E2D24A)	Data join - replicates template per datum
n	NestedJoin	Gold (#D4A017)	Type-changing data join
s	SceneJoin	Blue (#4A90E2)	GUP join with enter/update/exit
x	SceneNestedJoin	Purple (#9B4AE2)	GUP + type decomposition
a	Attr	Green (#4AE24A)	Attribute (child of Node)
b	Behavior	Orange (#E27A4A)	Behavior (child of Node)

Node Details

Node - The fundamental element node

- Has an ElementType (SVG, Group, Circle, etc.)
- Contains 0..n Attr children
- Contains 0..n Behavior children
- Contains 0..n Tree children
- Preserves phantom type `datum`

Join - Simple data join

- Replicates its template tree once per datum in the data array
- The template receives each datum
- Preserves phantom type `datum`

NestedJoin - Type-decomposing data join

- Like Join, but the decompose function extracts inner data
- **Changes phantom type**: `outerDatum -> innerDatum`
- Enables patterns like: `Array Row -> Array Cell`

SceneJoin - General Update Pattern join

- Handles enter/update/exit phases declaratively
- Each phase has optional initial attrs and transition config
- Preserves phantom type `datum`

SceneNestedJoin - GUP + type decomposition

- Combines NestedJoin's type changing with SceneJoin's GUP
- **Changes phantom type**: `outerDatum -> innerDatum`
- Recommended for most dynamic visualizations

ElementType Sub-menu

After pressing `e`, select the element type:

Key	ElementType	SVG Tag	Notes
`s`	SVG	`<svg>`	Root container
`g`	Group	`<g>`	Grouping element
`c`	Circle	`<circle>`	
`r`	Rect	`<rect>`	
`p`	Path	`<path>`	
`l`	Line	`<line>`	
`t`	Text	`<text>`	
`d`	Defs	`<defs>`	For gradients, patterns

Attribute Grammar

Attributes are children of Node elements.

```
Attr =
  | StaticAttr AttrName AttrValue
  | DataAttr AttrName AttrSource (datum -> AttrValue)
  | IndexedAttr AttrName AttrSource (datum -> Int -> AttrValue)
```

```
AttrValue = StringValue String | NumberValue Number | BooleanValue Boolean

AttrSource =
  | UnknownSource      -- Source not tracked
  | StaticSource String -- Literal value as string
  | FieldSource String  -- Field access: d.fieldName
  | ExprSource String   -- Arbitrary expression
  | IndexSource         -- Uses index parameter
```

AttrName Sub-menu

After pressing `a`, select the attribute name:

Key	AttrName	Applies to	Notes
`x`	x / cx	Rect, Circle	Horizontal position
`y`	y / cy	Rect, Circle	Vertical position
`w`	width	SVG, Rect	Width
`h`	height	SVG, Rect	Height
`r`	r / radius	Circle	Radius
`f`	fill	All shapes	Fill color
`s`	stroke	All shapes	Stroke color
`k`	stroke-width	All shapes	Stroke thickness
`o`	opacity	All	Opacity (0-1)
`d`	d	Path	Path data string
`t`	transform	All	Transform attribute
`c`	class	All	CSS class

AttrValue Sub-menu

After selecting AttrName, choose how the value is computed:

Key	Value Type	Color	Description
`l`	Lit (static)	Light blue (#93C5FD)	Fixed value
`f`	Field	Light green (#86EFAC)	`d.fieldName`
`e`	Expr	Light purple (#C4B5FD)	Arbitrary expression
`i`	Index	Light orange (#FED7AA)	Uses element index

Lit (Static) Values

For `l` (Lit), enter a literal value:

- Numbers: `100`, `3.14`, `-5`
- Strings: `"blue"`, `"#FF0000"`, `"translate(10,20)"`
- Booleans: `true`, `false`

Field Access

For `f` (Field), select or enter field name:

- Common fields: `x`, `y`, `value`, `name`, `color`, `size`
- Generates: `d.fieldName`

Expression

For `e` (Expr), enter arbitrary PureScript expression:

- `d.x + 10.0`
- `if d.active then "red" else "gray"`
- `show d.index`

Behavior Grammar

Behaviors are children of Node elements.

```
Behavior datum =
  | Zoom ZoomConfig
  | Drag DragConfig
  | ClickWithDatum (datum -> Effect Unit)
  | HoverWithDatum { enter: datum -> Effect Unit, leave: datum -> Effect
Unit }
```

Behavior Sub-menu

After pressing `b`, select the behavior type:

Key	Behavior	Color	Description
`z`	Zoom	Teal (#14B8A6)	Pan and zoom
`d`	Drag	Pink (#EC4899)	Draggable elements
`c`	Click	Red (#EF4444)	Click handler
`h`	Hover	Amber (#F59E0B)	Hover enter/leave

Phantom Type Flow

The phantom type `datum` flows through the tree:

```
Tree datum
  |
  +-- Node: preserves datum
  |     +-- children: Tree datum
  |
  +-- Join: preserves datum
  |     +-- template: datum -> Tree datum
  |
  +-- NestedJoin: CHANGES type
  |     +-- decompose: outerDatum -> Array innerDatum
  |     +-- template: innerDatum -> Tree innerDatum
  |
  +-- SceneJoin: preserves datum
  |     +-- template: datum -> Tree datum
  |
  +-- SceneNestedJoin: CHANGES type
        +-- decompose: outerDatum -> Array innerDatum
        +-- template: innerDatum -> Tree innerDatum
```

Type-changing nodes (NestedJoin, SceneNestedJoin) are visually marked to indicate the phantom type boundary.

Keyboard Navigation

Tree Navigation

- `ArrowUp` - Move to parent
- `ArrowDown` - Move to first child
- `ArrowLeft` - Move to previous sibling
- `ArrowRight` - Move to next sibling
- `Enter` - Confirm selection (in menus)
- `Escape` - Cancel / close menu

Node Creation

- `e` - Add Element node (opens ElementType menu)
- `j` - Add Join node
- `n` - Add NestedJoin node
- `s` - Add SceneJoin node
- `x` - Add SceneNestedJoin node
- `a` - Add Attribute (opens AttrName menu)
- `b` - Add Behavior (opens Behavior menu)

Menu Navigation

- `ArrowUp/Down` - Navigate menu items
- `Enter` - Select item
- `Escape` - Close menu
- Letter keys - Quick select by key

Visual Representation

Each node type has a distinct color for easy identification:

```
Node (Elem)      : Gray    #6B7280
Join             : Yellow  #E2D24A
NestedJoin       : Gold    #D4A017
SceneJoin        : Blue    #4A90E2
SceneNestedJoin  : Purple  #9B4AE2
Attr             : Green   #4AE24A
Behavior         : Orange  #E27A4A
```

Attr value types have lighter variants:

```
Lit (static)     : Light blue   #93C5FD
Field            : Light green  #86EFAC
Expr             : Light purple #C4B5FD
Index            : Light orange #FED7AA
```

Behavior types:

```
Zoom             : Teal   #14B8A6
Drag             : Pink   #EC4899
Click            : Red    #EF4444
Hover            : Amber  #F59E0B
```

Example Trees

Simple Bar Chart

```
Node SVG [width 800, height 600]
  +-- Node Group [class "bars"]
        +-- Join "bars" "rect" data
              +-- template: \d ->
                    Node Rect [x d.x, y d.y, width 20, height d.value,
fill "steelblue"]
```

Animated Circles with GUP

```
Node SVG [width 800, height 600]
  +-- SceneNestedJoin "circles" "circle" [sceneData] _.points
        +-- template: \point ->
              Node Circle [cx point.x, cy point.y, r 5]
```

```
          +-- enterBehavior: { initialAttrs: [r 0], transition: fadeIn }
          +-- updateBehavior: { attrs: [], transition: move }
          +-- exitBehavior: { attrs: [], transition: fadeOut }
```