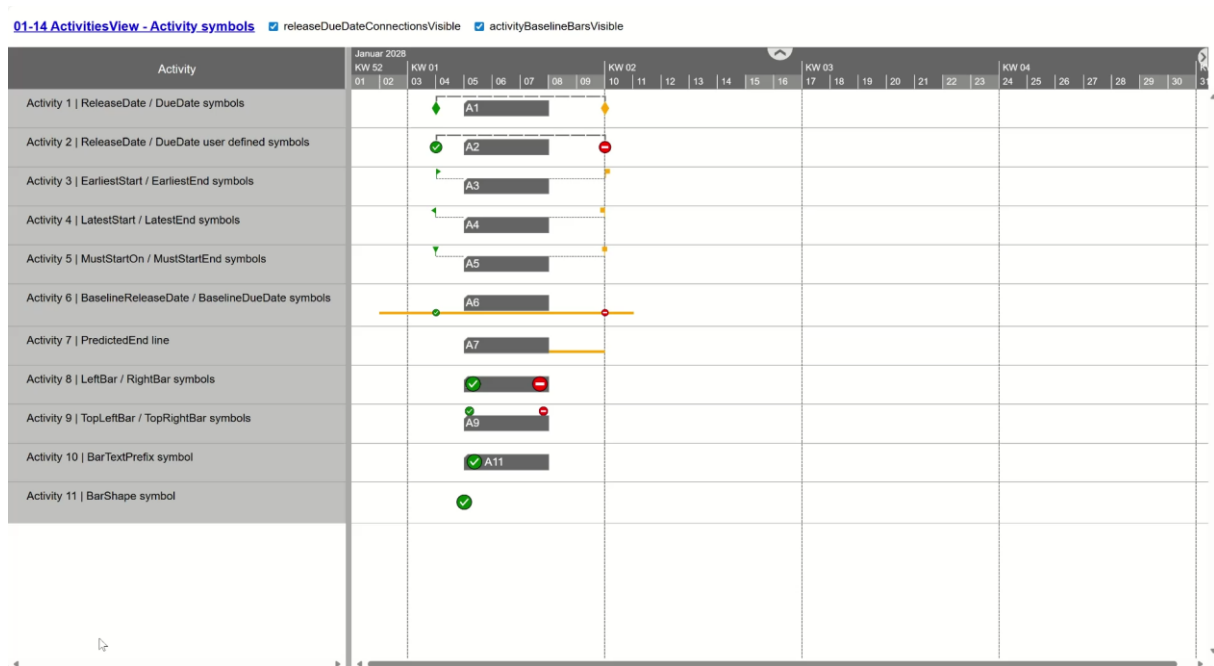


## 11-02 Activities view – Visualization - Symbols

This example shows which different symbols can be attached to the bars.



The release and due date symbols can be moved if the respective properties are set for the activity.

```
ReleaseDateAllowedDragModes: BarDragModes.DragHorizontally,
DueDateAllowedDragModes: BarDragModes.DragHorizontally,
```

If no calendar is defined, the option

```
ignoreCalendarOnActivityBarInteractions: true,
```

must also be activated, so that the symbols can be moved.

For the move to lead to a permanent change, the **onDrop** callback must react to the move of the symbols and set the new release or due date.

```
onDrop: function (args) {
    const activity = args.object;

    if (args.startPropertyName == 'ReleaseDate' && args.endPropertyName == undefined)
        activity.ReleaseDate = args.newStart;

    if (args.startPropertyName == 'DueDate' && args.endPropertyName == undefined)
        activity.DueDate = args.newEnd;

    vsWidget.updateActivities([activity]);
    vsWidget.render();
}
```

To ensure that the symbols for **EarliestStartDate**, **LatestStartDate** and **MustStartOnDate** are completely within the row, the **topRowMarginInTimeArea** option must be increased from 10 to 17 pixels.

```
topRowMarginInTimeArea: 17,
```

To maintain the centred position of the bar, it is recommended to set the **bottomRowMarginInTimeArea** option to the same value.

```
bottomRowMarginInTimeArea: 17,
```

The baseline consists of a bar to which symbols can be attached at any position. The baseline therefore has four date values.

```
BaselineStart: new Date(2028, 0, 4),
BaselineEnd: new Date(2028, 0, 10),
BaselineReleaseDate: new Date(2028, 0, 4),
BaselineDueDate: new Date(2028, 0, 10),
```

	Refers to	DateSource	SymbolID	Color	Height	Width	X-Offset
1.	ReleaseDate*	ReleaseDate					center
2.	DueDate*	DueDate					center
3.	EarliestStartDate	EarliestStart					right
4.	EarliestEndDate	EarliestEnd					right
5.	LatestStartDate	LatestStart					left
6.	LatestEndDate	LatestEnd					left
7.	MustStartOnDate	MustStartOn					center
8.	MustEndOnDate	MustStartOn					center
9.	BaselineStartDate	BaselineStart	No symbol.				center
10.	BaselineEndDate	BaselineEnd	No symbol.				center
11.	BaselineReleaseDate*	BaselineReleaseDate					center
12.	BaselineDueDate*	BaselineDueDate					center
13.	PredictedEndDate	PredictedEnd	No symbol.				center
14.	LeftBarSymbol	Start					right + Δ
15.	RightBarSymbol	End					left - Δ
16.	TopLeftBarSymbol	Start					right
17.	TopRightBarSymbol	End					left
18.	BarTextPrefixSymbol	Start					right + Δ
19.	BarShapeSymbol	Start					center

\*Colours can only be selected for the built-in Diamond symbol, but not for self-defined symbols.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>01-14 NETRONIC VSW SE</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <link href="../../VSWidget/nwaf-apptools.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-table.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-gantt.min.css" rel="stylesheet"/>
  <link href="../../VSWidget/nwaf-rab.min.css" rel="stylesheet"/>
  <script src="https://cdn.jsdelivr.net/npm/hammerjs@2.0.8/hammer.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/d3@7.9.0/dist/d3.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/tinycolor2@1.6.0/cjs/tinycolor.min.js"></script>
  <script src="../../VSWidget/nwaf-apptools.min.js"></script>
  <script src="../../VSWidget/nwaf-table.min.js"></script>
  <script src="../../VSWidget/nwaf-gantt.min.js"></script>
```

```

<script src="../../VSWidget/nwaf-rab.min.js"></script>
<script src="../../LicenseKey.js"></script>
<script src="Miscellaneous.js"></script>
</head>
<body>
  <div id="toolbar">
    <input id="connectionsVisible" type="checkbox" style="margin-left: 15px" checked/>
    <label for="connectionsVisible">releaseDueDateConnectionsVisible</label>
    <input id="baselineVisible" type="checkbox" style="margin-left: 15px" checked/>
    <label for="baselineVisible">activityBaselineBarsVisible</label>
  </div>
  <div id="VSWidget"></div>
  <script>
    Miscellaneous.ready(() => {
      const { BarShape, BarSortMode, BarDragModes, RowDesigns } = netronic.nVSW;

      document.getElementById("connectionsVisible").addEventListener("click", changeBarDesign);
      document.getElementById("baselineVisible").addEventListener("click", changeBarDesign);

      const timeAreaStart = new Date(2028, 0, 1);
      const timeAreaEnd = new Date(2028, 1, 1);

      const options = {
        licenseKey: window.VSW_LICENSE_KEY,

        start: timeAreaStart,
        end: timeAreaEnd,

        defaultActivityExpandedRowDesign: RowDesigns.Bars,
        defaultActivityCollapsedRowDesign: RowDesigns.Bars |
          RowDesigns.BarsInHiddenDescendantRows |
          RowDesigns.BarsStacked,
        activityBarSortModeForStackedRowDesign: BarSortMode.StartAndEnd,

        defaultActivityAllowedBarDragModes: BarDragModes.None,
        defaultActivityBarSelectable: false,
        defaultActivityRowSelectable: false,
        multipleSelectionEnabled: false,
        defaultActivityTableRowDefinitionID: "CustomRowDef",
        releaseDueDateConnectionsVisible: true,
        activityBaselineBarsVisible: true,
        ignoreCalendarOnActivityBarInteractions: true,
        topRowMarginInTimeArea: 17,
        bottomRowMarginInTimeArea: 17
      };

      const vsWidget = Miscellaneous.instantiateVSWidget(document.querySelector("#VSWidget"), options);

      const tableRowDefinitions = createTableRowDefinitions();
      const symbols = createSymbols();
      const activities = createActivities();
      vsWidget.addSymbols(symbols);
      vsWidget.addTableRowDefinitions(tableRowDefinitions);
      vsWidget.addActivities(activities);
      vsWidget.render();

      function createTableRowDefinitions() {
        const tableRowDefinitions = [
          {
            ID: "CustomRowDef",
            CellDefinitions: [
              {
                TitleText: "Activity",
                Width: 480,
              },
            ],
          },
        ],
      };
      return tableRowDefinitions;
    }

    function createSymbols() {
      const toSvgDataUri = (svg) => "data:image/svg+xml;utf8," + encodeURIComponent(svg.trim());
      const symbols = [

```

```

{
  id: "SymbolWarning",
  svg: `
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">
      <polygon points="12,2 22,20 2,20" fill="#f4b400" stroke="#7a5a00" stroke-width="1.5"/>
      <rect x="11" y="8" width="2" height="6" rx="1" fill="#1f1f1f"/>
      <circle cx="12" cy="17" r="1.4" fill="#1f1f1f"/>
    </svg>
  `,
},
{
  id: "SymbolCheckmark",
  svg: `
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">
      <circle cx="12" cy="12" r="10" fill="#34a853"/>
      <path d="M7 12.5l3.2 3.2l17 9" fill="none" stroke="black" stroke-width="2.4" stroke-linecap="round" stroke-linejoin="round"/>
    </svg>
  `,
},
{
  id: "SymbolOneway",
  svg: `
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">
      <circle cx="12" cy="12" r="10" fill="red" stroke="black" stroke-width="1"/>
      <rect x="6" y="10" width="12" height="4" fill="white"/>
    </svg>
  `,
},
{
  id: "SymbolDocument",
  svg: `
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">
      <path d="M7 2.75h7l5 5v20a1.25 1.25 0 0 1-1.25 1.25h-10.5A1.25 1.25 0 0 1 6 20v4A1.25 1.25 0 0 1 7.25 2.75z" fill="black" stroke="black" stroke-width="1.5"/>
      <path d="M14 2.75V8h5" fill="black" stroke="black" stroke-width="1.5" stroke-linejoin="round"/>
      <line x1="8.5" y1="11" x2="15.5" y2="11" stroke="black" stroke-width="1.4" stroke-linecap="round"/>
      <line x1="8.5" y1="14" x2="15.5" y2="14" stroke="black" stroke-width="1.4" stroke-linecap="round"/>
    </svg>
  `,
},
{
  id: "SymbolOpened",
  svg: `
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">
      <path d="M3 8.5A2.5 2.5 0 0 1 5.5 6h10l1.5 2h18.5A2.5 2.5 0 0 1 21 10.5V17a2.5 2.5 0 0 1-2.5 2.5h-13A2.5 2.5 0 0 1 3 17z" fill="black" stroke="black" stroke-width="1.4"/>
      <path d="M3.5 10h17l-1.7 7.2a1.5 1.5 0 0 1-1.46 1.15H6.1a1.5 1.5 0 0 1-1.47-1.18z" fill="black" stroke="black" stroke-width="1.2"/>
    </svg>
  `,
},
{
  id: "SymbolClosed",
  svg: `
    <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24">
      <rect x="6" y="10" width="12" height="10" rx="2" fill="gray" stroke="black" stroke-width="1"/>
      <path d="M8 10v7a4 4 0 1 1 8 0v3" stroke="black" stroke-width="1" fill="none"/>
      <circle cx="12" cy="15" r="1.5" fill="black"/>
      <line x1="12" y1="16.5" x2="12" y2="18" stroke="black" stroke-width="1"/>
    </svg>
  `,
}
};

```

```

    return symbols.map(s => ({
      ID: s.id,
      URL: toSvgDataUri(s.svg)
    }));
  }

function createActivities() {
  const activities = [
    {
      ID: "Act1",
      TableText: "Activity 1 | ReleaseDate / DueDate symbols",
      BarText: "A1",
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 8),
      ReleaseDate: new Date(2028, 0, 4),
      ReleaseDateColor: "green",
      DueDate: new Date(2028, 0, 10),
      DueDateColor: "orange",
      ReleaseDateAllowedDragModes: BarDragModes.DragHorizontally,
      DueDateAllowedDragModes: BarDragModes.DragHorizontally,
    },
    {
      ID: "Act2",
      TableText: "Activity 2 | ReleaseDate / DueDate user defined symbols",
      BarText: "A2",
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 8),
      ReleaseDate: new Date(2028, 0, 4),
      ReleaseDateColor: "green",
      ReleaseDateSymbolID: "SymbolCheckmark",
      ReleaseDateSymbolHeight: 20,
      ReleaseDateSymbolWidth: 20,
      DueDate: new Date(2028, 0, 10),
      DueDateSymbolID: "SymbolOneway",
      DueDateSymbolHeight: 20,
      DueDateSymbolWidth: 20,
      DueDateColor: "orange",
      ReleaseDateAllowedDragModes: BarDragModes.DragHorizontally,
      DueDateAllowedDragModes: BarDragModes.DragHorizontally,
    },
    {
      ID: "Act3",
      TableText: "Activity 3 | EarliestStart / EarliestEnd symbols",
      BarText: "A3",
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 8),
      EarliestStart: new Date(2028, 0, 4),
      EarliestStartColor: "green",
      EarliestEnd: new Date(2028, 0, 10),
      EarliestEndColor: "orange",
    },
    {
      ID: "Act4",
      TableText: "Activity 4 | LatestStart / LatestEnd symbols",
      BarText: "A4",
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 8),
      LatestStart: new Date(2028, 0, 4),
      LatestStartColor: "green",
      LatestEnd: new Date(2028, 0, 10),
      LatestEndColor: "orange",
    },
    {
      ID: "Act5",
      TableText: "Activity 5 | MustStartOn / MustStartEnd symbols",
      BarText: "A5",
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 8),
      MustStartOn: new Date(2028, 0, 4),
      MustStartOnColor: "green",
      MustEndOn: new Date(2028, 0, 10),
      MustEndOnColor: "orange",
    },
  ],
  {

```

```

ID: "Act6",
TableText: "Activity 6 | BaselineReleaseDate / BaselineDueDate symbols",
BarText: "A6",
Start: new Date(2028, 0, 5),
End: new Date(2028, 0, 8),
BaselineStart: new Date(2028, 0, 2),
BaselineEnd: new Date(2028, 0, 11),
BaselineReleaseDate: new Date(2028, 0, 4),
BaselineDueDate: new Date(2028, 0, 10),
BaselineColor: "orange",
BaselineReleaseDateColor: "green",
BaselineReleaseDateSymbolID: "SymbolCheckmark",
BaselineDueDateColor: "orange",
BaselineDueDateSymbolID: "SymbolOneway",
BaselineBorderColor: "orange",
},
{
ID: "Act7",
TableText: "Activity 7 | PredictedEnd line",
BarText: "A7",
Start: new Date(2028, 0, 5),
End: new Date(2028, 0, 8),
PredictedEnd: new Date(2028, 0, 10),
PredictedEndColor: "orange",
},
{
ID: "Act8",
TableText: "Activity 8 | LeftBar / RightBar symbols",
BarText: "A8",
Start: new Date(2028, 0, 5),
End: new Date(2028, 0, 8),
LeftBarSymbolID: "SymbolCheckmark",
LeftBarSymbolHeight: 25,
LeftBarSymbolWidth: 25,
RightBarSymbolID: "SymbolOneway",
RightBarSymbolHeight: 25,
RightBarSymbolWidth: 25,
},
{
ID: "Act9",
TableText: "Activity 9 | TopLeftBar / TopRightBar symbols",
BarText: "A9",
Start: new Date(2028, 0, 5),
End: new Date(2028, 0, 8),
TopLeftBarSymbolID: "SymbolCheckmark",
TopLeftBarSymbolHeight: 15,
TopLeftBarSymbolWidth: 15,
TopLeftBarSymbolYOffset: -4,
TopRightBarSymbolID: "SymbolOneway",
TopRightBarSymbolHeight: 15,
TopRightBarSymbolWidth: 15,
TopRightBarSymbolYOffset: -4,
},
{
ID: "Act10",
TableText: "Activity 10 | BarTextPrefix symbol",
BarText: "A11",
Start: new Date(2028, 0, 5),
End: new Date(2028, 0, 8),
BarTextPrefixSymbolID: "SymbolCheckmark",
BarTextPrefixSymbolHeight: 25,
BarTextPrefixSymbolWidth: 25,
},
{
ID: "Act11",
TableText: "Activity 11 | BarShape symbol",
BarText: "A11",
Start: new Date(2028, 0, 5),
End: new Date(2028, 0, 8),
BarShape: BarShape.Symbol,
BarShapeSymbolID: "SymbolCheckmark",
BarHeight: 25,
BarShapeSymbolWidth: 25,
},

```

```
    ];  
  
    return activities;  
  }  
  
  function changeBarDesign(event) {  
    const checkbox = event.target;  
  
    if (checkbox.id === "connectionsVisible")  
      vsWidget.option("releaseDueDateConnectionsVisible", checkbox.checked);  
    else if (checkbox.id === "baselineVisible")  
      vsWidget.option("activityBaselineBarsVisible", checkbox.checked);  
  }  
});  
</script>  
</body>  
</html>
```