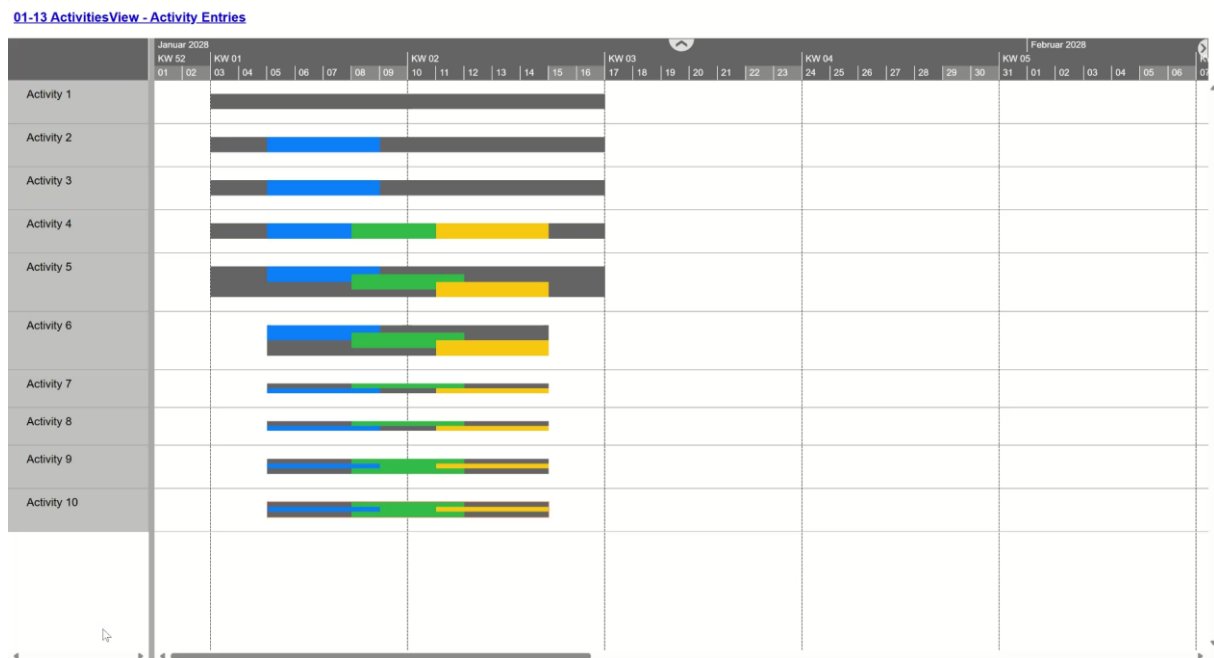


## 10-03 Activities view – Basics - Activity entries

This example shows how activity entries are defined and what you need to know when dealing with activity entries.



**Activity 1** is a normal activity that does not yet contain any entries.

**Activity 2** has been defined in the same way as Activity 1, but contains an additional entry (blue bar).

**Activity 3** was defined in the same way as Activity 1, but the activity bar contains an additional Color entry. This is why the activity bar is visible again.

**Activity 4** contains two further entries. Entries are displayed in the specified order.

**Activity 5** makes the overlaps that exist in Activity 4 visible by specifying a suitable *RelativeTopOffset* for each entry. The height of the activity bar is extended to the total height via the *Height* property.

**Activity 6** no longer contains values for the *Start* and *End* properties of the activity bar. The time span is now determined by the entries in *Entries*. The following general rule applies: A specified *Start* value is only considered if it is earlier than the minimum of all entry *Start* dates. The *End* property only takes effect if the specified end date is later than the maximum of all entries.

**Activity 7** shows a variation in the height and arrangement of the entry bars by setting the *Height* property and adjusting *RelativeTopOffset*.

**Activity 8** highlights the bars of the entries because the color for the activity bar has been omitted.

**Activity 9** has a different order for entries to ensure that the green bar with a greater height lies behind the narrower bars.

**Activity 10** has been given a visible border due to the *BorderWidth* and *BorderColor* settings. In certain application scenarios, the border can be used effectively to indicate the coherence of the individual entries.

The activity bars with their entries can be moved if the two options are set:

```
defaultActivityAllowedBarDragModes: BarDragModes.DragHorizontally,  
ignoreCalendarOnActivityBarInteractions: true,
```

In this example, the Property *ignoreCalendarOnActivityBarInteractions* must be set because no calendar is defined.

To update the underlying values at the end of an interactive dragging, the *OnDrop* callback must handle this accordingly. The *Start* and *End* values need to be updated. If this is not done, the bars return to their original position.

```
onDrop:  
  function (args) {  
    const activity = args.object;  
  
    if (activity.Start !== undefined)  
      activity.Start = new Date(activity.Start.getTime() + args.workingTimeDistance);  
  
    if (activity.End !== undefined)  
      activity.End = new Date(activity.End.getTime() + args.workingTimeDistance);  
  
    if (activity.Entries !== undefined) {  
      for (let i = 0; i < activity.Entries.length; i++) {  
        activity.Entries[i].Start = new Date(activity.Entries[i].Start.getTime() +  
          args.workingTimeDistance);  
        activity.Entries[i].End = new Date(activity.Entries[i].End.getTime() +  
          args.workingTimeDistance);  
      }  
    }  
    vsWidget.updateActivities([activity]);  
    vsWidget.render();  
  }  
}  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <title>01-13 NETRONIC VSW SE</title>  
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
  <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">  
    <h3>  
      <a href="docs/01-13_ActivitiesView_ActivityEntries.pdf" target="_blank">01-13 ActivitiesView - Activity Entries</a>  
    </h3>  
  </div>  
</body>  
</html>
```

```

</div>
<div id="VSWidget"></div>
<script>
    Miscellaneous.ready(() => {
        const { ActivityBarShape, BarDragModes, RowDesigns } = netronic.nVSW;

        const timeAreaStart = new Date(2028, 0, 1);
        const timeAreaEnd = new Date(2028, 3, 1);

        const options = {
            licenseKey: window.VSW_LICENSE_KEY,

            start: timeAreaStart,
            end: timeAreaEnd,

            defaultActivityExpandedRowDesign: RowDesigns.Bars,
            defaultActivityAllowedBarDragModes: BarDragModes.DragHorizontally,
            defaultActivityBarSelectable: false,
            defaultActivityRowSelectable: false,
            multipleSelectionEnabled: false,
            topRowMarginInTimeArea: 20,
            bottomRowMarginInTimeArea: 20,
            ignoreCalendarOnActivityBarInteractions: true,

            onDrop: (args) => {
                const activity = args.object;
                if (activity.Start)
                    activity.Start = new Date(activity.Start.getTime() + args.workingTimeDistance);

                if (activity.End)
                    activity.End = new Date(activity.End.getTime() + args.workingTimeDistance);

                if (activity.Entries) {
                    for (let i = 0; i < activity.Entries.length; i++) {
                        activity.Entries[i].Start = new Date(activity.Entries[i].Start.getTime() +
                            args.workingTimeDistance);
                        activity.Entries[i].End = new Date(activity.Entries[i].End.getTime() +
                            args.workingTimeDistance);
                    }
                }
                vsWidget.updateActivities([activity]);
                vsWidget.render();
            },
        };

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

        const activities = createActivities();
        vsWidget.addActivities(activities);
        vsWidget.render();

        function createActivities() {
            const activities = [
                {
                    ID: "Act1",
                    TableText: "Activity 1",
                    Start: new Date(2028, 0, 3),
                    End: new Date(2028, 0, 17),
                    BarShape: ActivityBarShape.Rectangle,
                    BorderWidth: 0,
                },
                {
                    ID: "Act2",
                    TableText: "Activity 2",
                    Start: new Date(2028, 0, 3),
                    End: new Date(2028, 0, 17),
                    BarShape: ActivityBarShape.Rectangle,
                    BorderWidth: 0,
                    Entries: [
                        {
                            Start: new Date(2028, 0, 5),
                            End: new Date(2028, 0, 9),
                            Color: "#007bff",
                            Height: 22,
                        }
                    ]
                }
            ];
        }
    });

```

```

    },
  ],
},
{
  ID: "Act3",
  TableText: "Activity 3",
  Start: new Date(2028, 0, 3),
  End: new Date(2028, 0, 17),
  BarShape: ActivityBarShape.Rectangle,
  BorderWidth: 0,
  Color: "#646464",
  Entries: [
    {
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 9),
      Color: "#007bff",
      RelativeTopOffset: 0,
      Height: 22,
    },
  ],
},
},
{
  ID: "Act4",
  TableText: "Activity 4",
  Start: new Date(2028, 0, 3),
  End: new Date(2028, 0, 17),
  BarShape: ActivityBarShape.Rectangle,
  Color: "#646464",
  BorderWidth: 0,
  Entries: [
    {
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 9),
      Color: "#007bff",
    },
    {
      Start: new Date(2028, 0, 8),
      End: new Date(2028, 0, 12),
      Color: "#28a745",
    },
    {
      Start: new Date(2028, 0, 11),
      End: new Date(2028, 0, 15),
      Color: "#ffc107",
    },
  ],
},
},
{
  ID: "Act5",
  TableText: "Activity 5",
  Start: new Date(2028, 0, 3),
  End: new Date(2028, 0, 17),
  BarShape: ActivityBarShape.Rectangle,
  BarHeight: 44,
  Color: "#646464",
  BorderWidth: 0,
  BarTopOffset: 0,
  Entries: [
    {
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 9),
      Color: "#007bff",
      RelativeTopOffset: 0,
      Height: 22,
    },
    {
      Start: new Date(2028, 0, 8),
      End: new Date(2028, 0, 12),
      Color: "#28a745",
      Height: 22,
      RelativeTopOffset: 11,
    },
    {
      Start: new Date(2028, 0, 11),

```

```

        End: new Date(2028, 0, 15),
        Color: "#ffc107",
        RelativeTopOffset: 22,
        Height: 22,
    },
],
},
{
    ID: "Act6",
    TableText: "Activity 6",
    BarShape: ActivityBarShape.Rectangle,
    BarHeight: 44,
    Color: "#646464",
    BorderWidth: 0,
    BarTopOffset: 0,
    Entries: [
        {
            Start: new Date(2028, 0, 5),
            End: new Date(2028, 0, 9),
            Color: "#007bff",
            RelativeTopOffset: 0,
            Height: 22,
        },
        {
            Start: new Date(2028, 0, 8),
            End: new Date(2028, 0, 12),
            Color: "#28a745",
            Height: 22,
            RelativeTopOffset: 11,
        },
        {
            Start: new Date(2028, 0, 11),
            End: new Date(2028, 0, 15),
            Color: "#ffc107",
            RelativeTopOffset: 22,
            Height: 22,
        },
    ],
},
{
    ID: "Act7",
    TableText: "Activity 7",
    BarShape: ActivityBarShape.Rectangle,
    Color: "#646464",
    BorderWidth: 0,
    BarHeight: 14,
    Entries: [
        {
            Start: new Date(2028, 0, 5),
            End: new Date(2028, 0, 9),
            Color: "#007bff",
            Height: 7,
            RelativeTopOffset: 7,
        },
        {
            Start: new Date(2028, 0, 8),
            End: new Date(2028, 0, 12),
            Color: "#28a745",
            Height: 7,
        },
        {
            Start: new Date(2028, 0, 11),
            End: new Date(2028, 0, 15),
            Color: "#ffc107",
            Height: 7,
            RelativeTopOffset: 7,
        },
    ],
},
{
    ID: "Act8",
    TableText: "Activity 8",
    BarShape: ActivityBarShape.Rectangle,
    BorderWidth: 0,

```

```

BarHeight: 14,
Entries: [
  {
    Start: new Date(2028, 0, 5),
    End: new Date(2028, 0, 9),
    Color: "#007bff",
    Height: 7,
    RelativeTopOffset: 7,
  },
  {
    Start: new Date(2028, 0, 8),
    End: new Date(2028, 0, 12),
    Color: "#28a745",
    Height: 7,
  },
  {
    Start: new Date(2028, 0, 11),
    End: new Date(2028, 0, 15),
    Color: "#ffc107",
    Height: 7,
    RelativeTopOffset: 7,
  },
],
},
{
  ID: "Act9",
  TableText: "Activity 9",
  BarShape: ActivityBarShape.Rectangle,
  BorderWidth: 0,
  Entries: [
    {
      Start: new Date(2028, 0, 8),
      End: new Date(2028, 0, 12),
      Color: "#28a745",
    },
    {
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 9),
      Color: "#007bff",
      Height: 7,
      RelativeTopOffset: 7,
    },
    {
      Start: new Date(2028, 0, 11),
      End: new Date(2028, 0, 15),
      Color: "#ffc107",
      Height: 7,
      RelativeTopOffset: 7,
    },
  ],
},
{
  ID: "Act10",
  TableText: "Activity 10",
  BarShape: ActivityBarShape.Rectangle,
  BorderWidth: 1,
  BorderColor: "coral",
  Entries: [
    {
      Start: new Date(2028, 0, 8),
      End: new Date(2028, 0, 12),
      Color: "#28a745",
    },
    {
      Start: new Date(2028, 0, 5),
      End: new Date(2028, 0, 9),
      Color: "#007bff",
      Height: 7,
      RelativeTopOffset: 7,
    },
    {
      Start: new Date(2028, 0, 11),
      End: new Date(2028, 0, 15),
      Color: "#ffc107",
    },
  ],
}

```

```
                Height: 7,  
                RelativeTopOffset: 7,  
            },  
        ],  
    ],  
    return activities;  
}  
});  
</script>  
</body>  
</html>
```