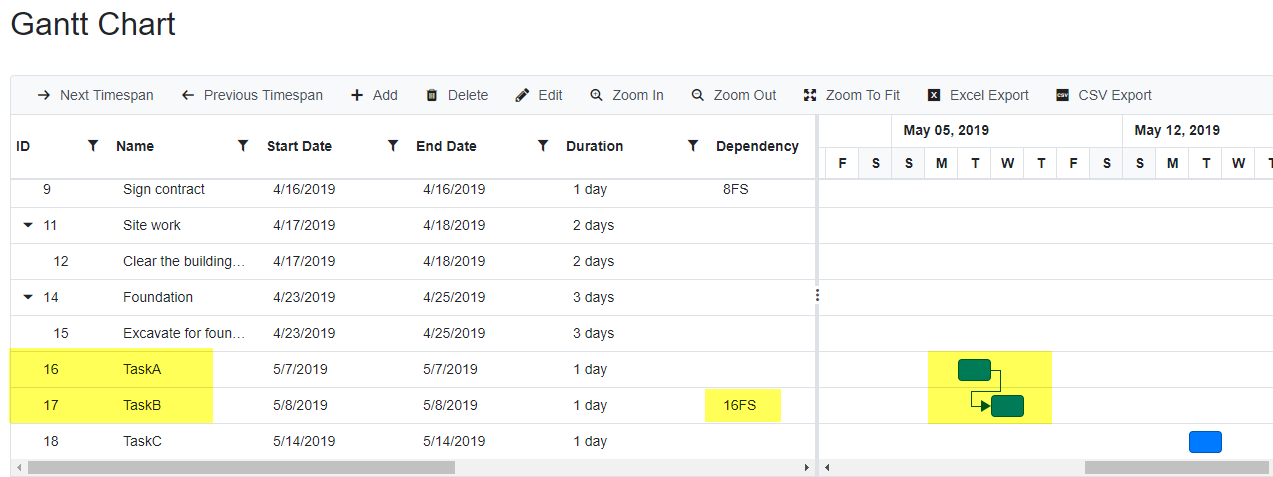
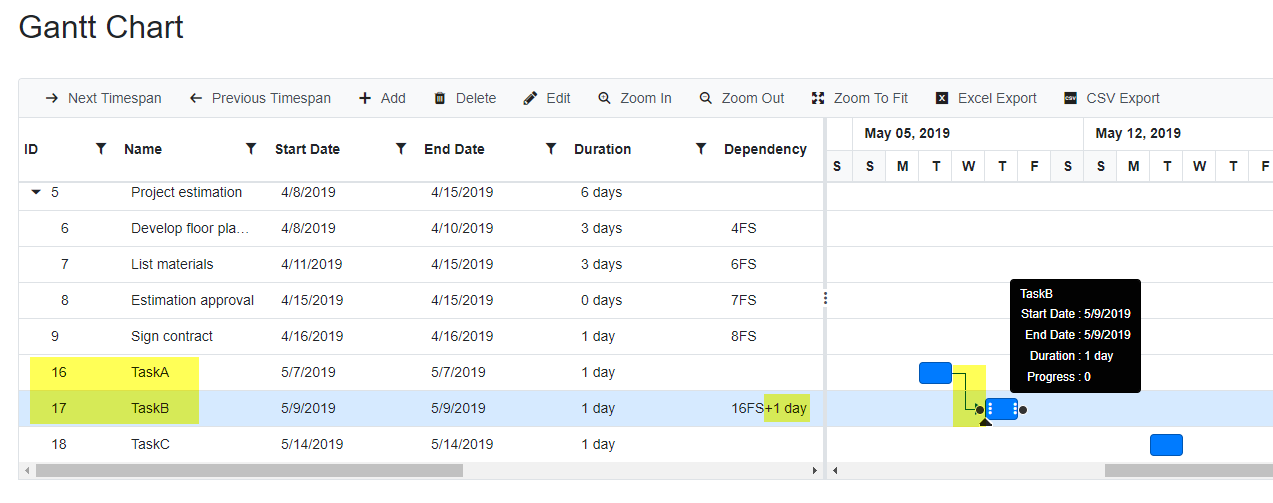
Currently, the **Syncfusion Blazor Gantt Chart control** treats dependency lags and leads as absolutes. Our company’s BST10 Gantt Chart, as well as, Microsoft Project’s Gantt Chart treats lags and leads as mandatory minimums. Let me demonstrate what I mean by this in the next few examples:

**Syncfusion Blazor Gantt Chart Control Behavior**

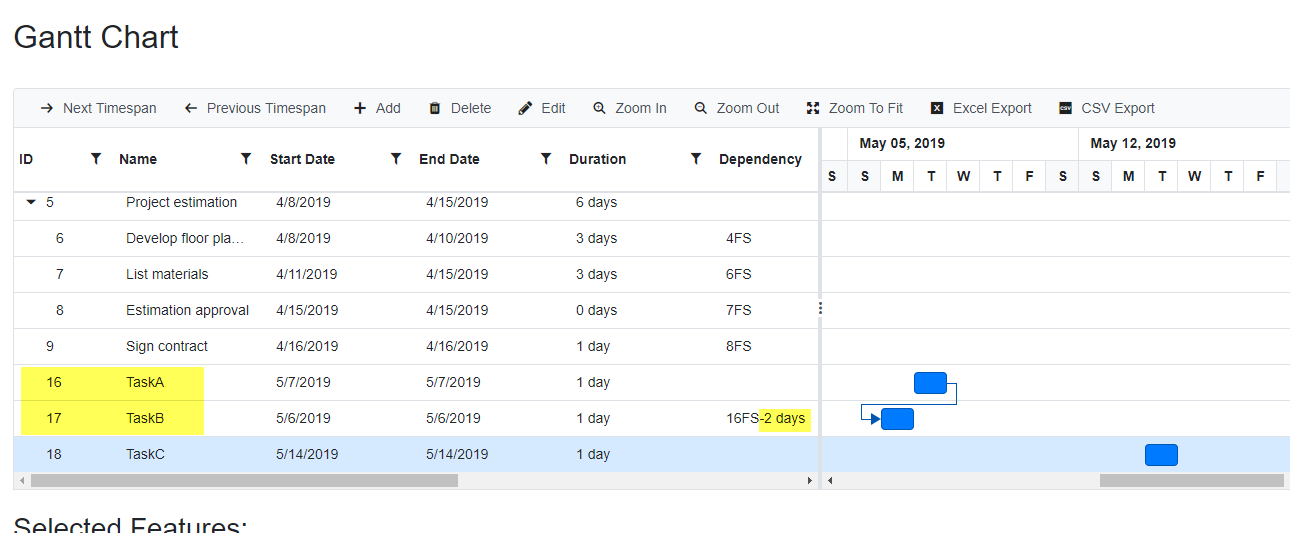
**A)** Here we have a very basic dependency between TaskA and TaskB. TaskB is the successor of TaskA. TaskA is the predecessor of TaskB.



**B)** If you drag TaskB out into the future a day, the following will be displayed:  
This creates a “lag” of 1 day.



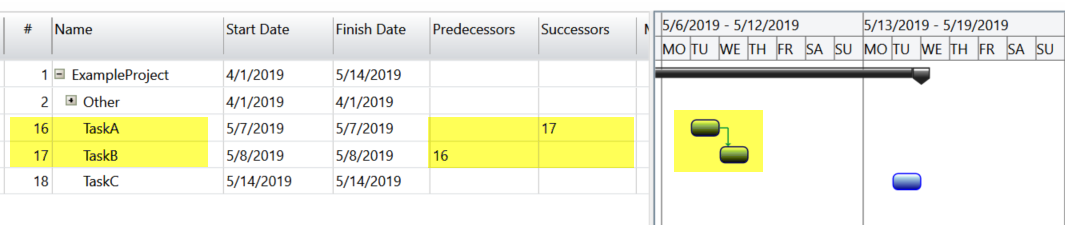
**C)** If instead, you drag TaskB out 2 days into the past, the following will be displayed:  
This creates a “lead” of 2 days.



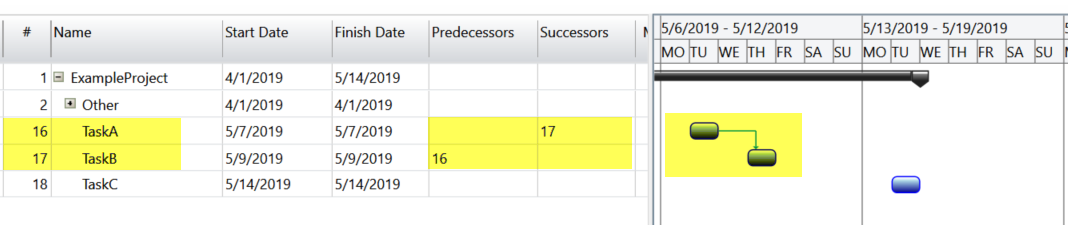
The same behaviors can be mimicked using the Dependency string property by typing in the desired dependency with lag or lead.

**Our Companies Gantt Chart Behavior**

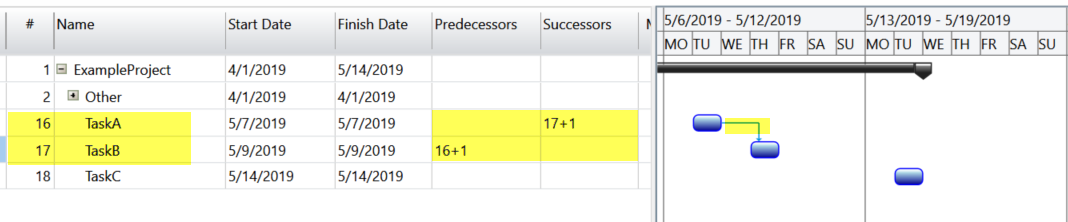
**D)** This is a replica of example A (shown above) using our BST10 Gantt Chart:

At this point, the user will NOT be able to drag TaskB into the past but will be allowed to drag TaskB into the future. Our control does not allow users to create dependency leads by dragging task bars but instead requires the user to add them via the Predecessor or Successor string properties. Our Predecessor field is the same as your Blazor Gantt controls dependency string property.  
  


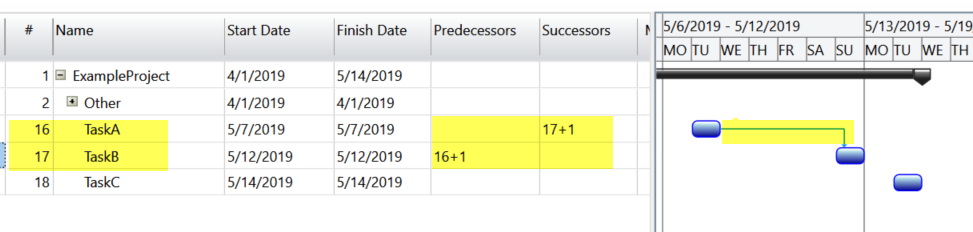
**E)** If you drag TaskB out into the future by a day, the following will be displayed:  
This is a replica of Example B but as you can see unlike in example B, our Predecessor string property (which is the same as your dependency string property) does not display a “+1 day”. This is because our BST10 Gantt control considers lag and lead to be ‘mandatory minimums’. This allows the user to drag tasks with dependencies into the future wherever they believe the task should/might start. The lag/lead acts more as a boundary. Let me try to explain this further with more examples.



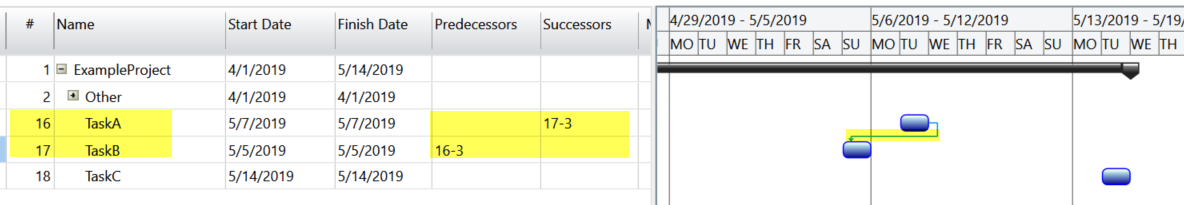
**F)** Here I have added a lag of 1 day by typing in our predecessor string property:  
At this point, a user will not be allowed to move TaskB into the past because the lag restricts it from moving any further backwards in time. But again, the user will be allowed to drag the task into the future without restriction.



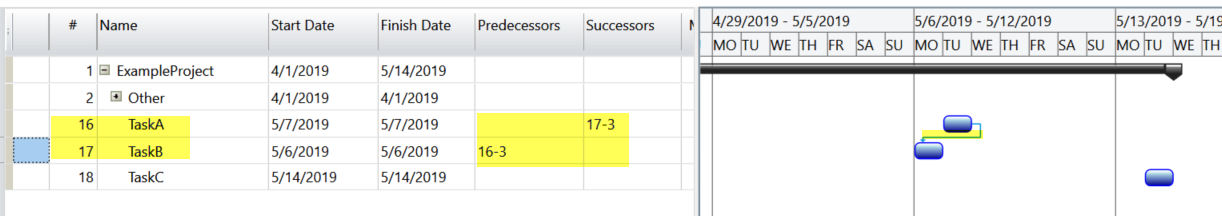
**G)** Here we are displaying a lag of 1 day on the dependency between TaskA and TaskB but the user has also decided to drag TaskB out into the future. This allows the user to know that as a minimum TaskB can start on 5/9/2019 and not on 5/8/2019 because of the lag created. But also allows the user to decide that TaskB will likely not start until 5/12/2019.



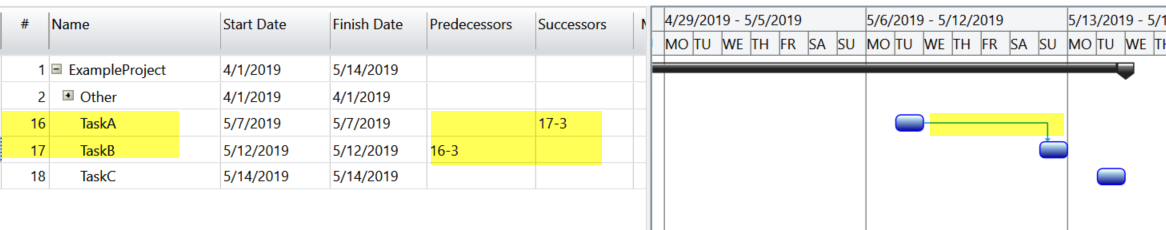
**H)** Here we have created a lead of 3 days for the dependency between TaskA and TaskB:  
At this point, TaskB cannot be dragged any further into the past because 5/8/2019 – 3 days is 5/5/2019. Which makes 5/5/2019 its mandatory minimum start date.



**I)** Here we’ve maintained the 3-day lead within the predecessor string property, for the dependency between TaskA and TaskB, but have dragged TaskB one day into the future:



**J)** Here we’ve maintained the 3-day lead within the predecessor string property, for the dependency between TaskA and TaskB, but we have dragged it several days into the future.



Will Syncfusion be updating the Gantt Charts concept of lag to be more like our BST10 Gantt Chart and Microsoft Projects Gantt Chart? Or will lags and leads continue to be treated like absolutes within the control?