

Visual Production Scheduler for Microsoft Dynamics NAV

How to... Use Indicators

(E.G. TO SHOW WARNINGS REGARDING MATERIAL AVAILABILITY ON PRODUCTION ORDER LINES OR OPERATIONS)

In addition to the **visual alerts** indicating **impending collisions of scheduled orders with operational processes** (red capacity curve for impending overload, hatching for delivery date exceeding and red underlining for overlapping operations) the VPS offers a feature for highlighting cases following **your own business rules by changing the main font color of the planning table**.

You can choose whether to modify the font color in the table area on Production Order Lines level and/or on operation level in the Gantt area. This is done by so-called **indicators** that have to be programmed externally. The color can be set by an ARGB value. The color value will be reduced internally to the nearest of 27 colors represented by all combinations of 0, 128 and 255 for the R, G and B values.

The following screenshot shows two indicators “at work”.

1. Indicator on production order line “1011002 – 10000” with font color red
2. Indicator on operation “20” of production order line “1011002 – 10000” with font color blue

The screenshot displays a production planning table with a tree view on the left and a Gantt chart on the right. The tree view shows a hierarchy: 101004 - 50000 (expanded), 1011002 - 10000 (expanded, with red font), and Assembly department (expanded). Under Assembly department, three resources are listed: Mike Seamans, Bryan Walton, and Linda Mitchell. The Gantt chart shows operations for these resources. Operation 10 is shown for Mike Seamans, operation 20 for Bryan Walton, and operation 30 for Linda Mitchell. The text '1011002 - 10000' and '20' are highlighted in red and blue respectively, indicating the active indicators.

▶ 101004 - 50000	
▲ 1011002 - 10000	
▲ Assembly department	10
Mike Seamans	
Bryan Walton	20
Linda Mitchell	30

DEFINITION OF AN INDICATOR OBJECT

To set an indicator, an indicator object has to be created and the according properties identifying the objects the indicator is to be applied to have to be filled. This indicator object has to be added to a list so that if you have defined more than one indicator they will be applied one after the other. The following table shows the settings needed for an indicator object:

<i>Indicator</i>		
<i>Property</i>	<i>Data type</i>	<i>Description</i>
IndicatorObjectType	Enum/Option	Specifies which object is identified by the following properties: 0 = Undefined 1 = View 2 = Status 3 = ProductionOrder 4 = ProductionOrderLine 5 = ProductionOrderRoutingLine 6 = WorkCenter 7 = MachineCenter
POStatus	String	
PONo	String	
POLineNo	String	
RoutingNo	String	
RoutingRefNo	String	
OPNo	String	
WCNo	String	
MCNo	String	
IndicatorValue	String	RGB color value
IndicatorDisplayType	Enum/Option	How ist he indicator to be displayed: 0 = Undefined 1 = MainFontColor
ValidityType	Enum/Option	How long is the indicator valid: 0 = Undefined 1 = Permanent 2 = FirstChangeOfDataInSchedule
ViewType	Enum/Option	View the indicator belongs to: 0 = Undefined 1 = ProductionOrders 2 = Capacities 4 = Loads

APPLYING INDICATORS AUTOMATICALLY AFTER THE VPS HAS BEEN STARTED.

In order to apply your indicators automatically after having started the VPS, you have to modify the event **OnDataLoaded**. This could be done like shown below:

```
HandleAddInMessage (Index : Integer; Data : BigText)
CASE Index OF

    // EVENTID_OnDataLoaded
    2100:
    BEGIN
        ProcessDataLoaded();
    END;
```

In our sample implementation, we have enhanced the event by the **ProcessDataLoaded()** function which calls the **SetAllIndicatorsForSchedule()** function to create indicators, equip them with values and handle them over to the VPS.

Sample code for creating an indicator

The following code snippet is taken from the **SetAllIndicatorsForSchedule()** code for creating an indicator for a „ProductionOrderRoutingLine“ (operation).

In this code snippet, the necessary values for identifying the correct RoutingLine and color and validity are specified.

```
Indicator.IndicatorObjectType :=
    IndicatorObjectType.ProductionOrderRoutingLine;
Indicator.POStatus :=
    FORMAT(ProdOrderRoutingLine.Status,0,2);
Indicator.PONo :=
    FORMAT(ProdOrderRoutingLine."Prod. Order No.");
Indicator.POLineNo :=
    FORMAT(lProdOrderLine."Line No.");
Indicator.RoutingNo :=
    FORMAT(ProdOrderRoutingLine."Routing No.");
Indicator.RoutingRefNo :=
    FORMAT( ProdOrderRoutingLine."Routing Reference No.");
Indicator.OPNo :=
    FORMAT(ProdOrderRoutingLine."Operation No.");
Indicator.IndicatorDisplayType :=
    IndicatorDisplayType.MainFontColor;
Indicator.IndicatorValue :=
    FORMAT(lColor.Yellow.ToArgb());
Indicator.IndicatorValidityType :=
    IndicatorValidityType.FirstChangeOfDataInSchedule;
```

For the complete code of the sample implementation, see the **SetAllIndicatorsForSchedule()** function.

APPLYING INDICATORS BY USER INTERACTION

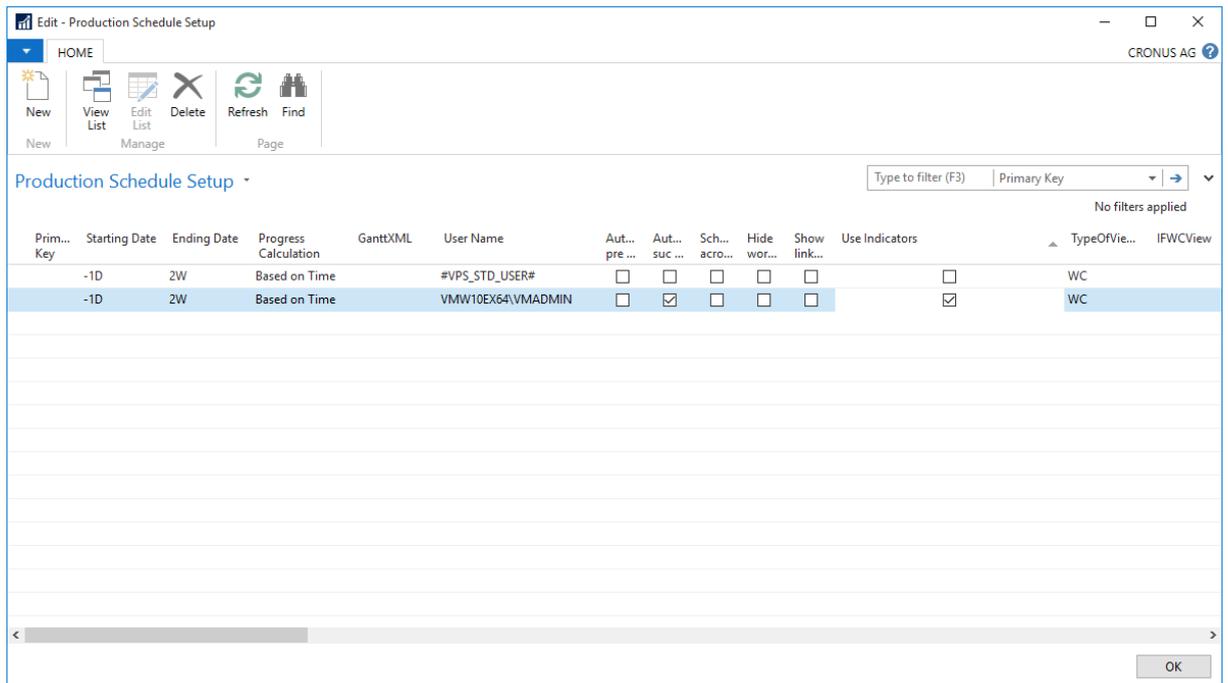
In addition to applying indicators automatically upon starting the VPS, the decision whether to use indicators or not and at which position can be left to the user by offering an additional item in the context menu. This way of using indicators bears the advantage of reducing complex calculations since the user only retrieves them for certain selected objects.

Two steps are needed to use the indicators:

1. Implement the **ProcessRequestIndicator**:

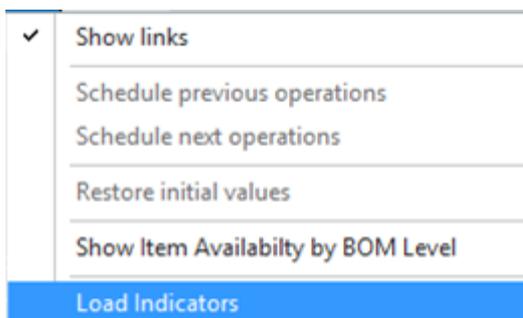
```
// EVENTID_RequestIndicators
2240:
BEGIN
    CLEAR(myBigText);
    ProdSchedMan.ProcessRequestIndicator(Data, myBigText);
    SetAddInValueByBT('1140',myBigText);
END;
```

2. Tick the “Use Indicators” check box in the NAV Table “NETRONIC VPS Setup” (5416000)



As a result, the item “Load indicators” or “Load and Save Indicators” respectively is added to the context menu:

Please note that you cannot load an indicator if there are any unsaved changes in the planning board. As a consequence the context menu item will change to “Save and Load indicators” if necessary so that the user has to save his changes before the indicators can be set.



Clicking this item starts a request, the request object being of the indicator object type and specifying the object for which the indicator was requested. This could be an operation in the diagram area, e.g., or a group in the table area. The object at which the option was clicked determines the value set in the **IndicatorObjectType** property.

The complete sample code can be found in the **ProcessRequestIndicator()** function