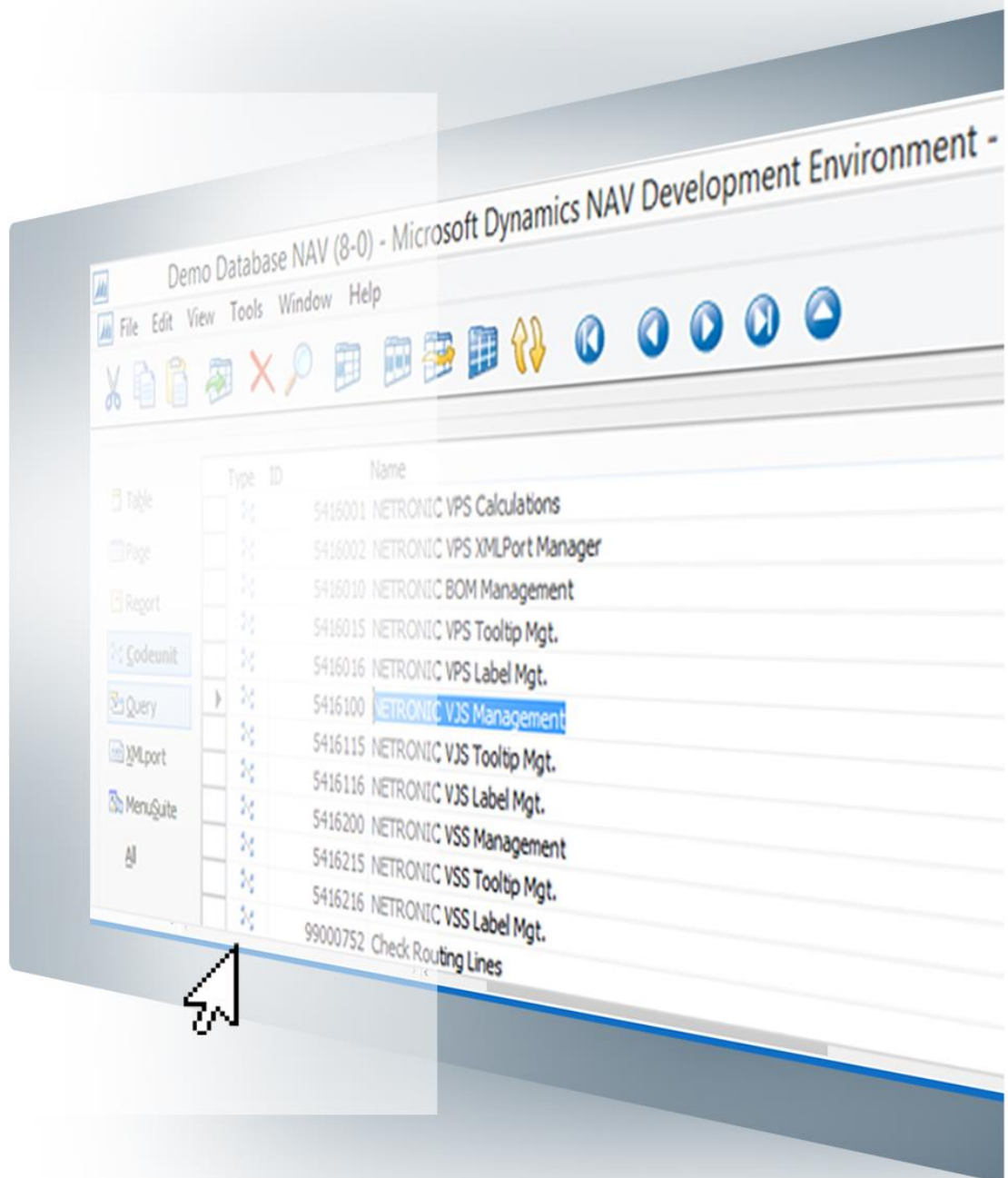


## TECHNICAL GUIDE FOR PARTNERS



## Enhancing the Visual Jobs Scheduler for Microsoft Dynamics NAV by C/AL Code Customizations

# Content

---

Preliminaries .....	3
Modify an existing menu suite (NAV object) in the Visual Scheduling Suite.....	4
Individual customization.....	4
Use filter settings of a NAV page for the VJS.....	5
VJS Version 1.1 and below.....	5
VJS Version 1.2 and above.....	6
Example 1 Only show project of project card.....	6
Example 2 – show projects from project list .....	7
Start the VJS in the Resource View .....	8
Use date and time fields for positioning job planning lines in the diagram.....	9
Step 1 .....	9
Step 2 .....	10
Step 3 .....	10
Step 4 .....	10
Use milestones.....	11
Milestones on Job Level.....	11
Activate the milestones .....	11
Result .....	11
Milestones on Job Task level .....	12
Activate the milestones .....	12
Result .....	12
Handle capacities for non-project tasks .....	13
How to proceed: .....	13
Modify the sections in codeunit 5416100 NETRONIC VJS Management: .....	14
Section 1.....	14
Section 2.....	14
Modify the sections in Page 5416102 NETRONIC VJS JPL Insert .....	14
Further Resources.....	15
About NETRONIC - the Gantt Company.....	15

June 2018

## Preliminaries

---

The VJS add-in for graphical project planning in Microsoft Dynamics NAV offers powerful and versatile planning functionality. However, sometimes customers or partners may need features that go way beyond the standard VJS installation.

This guide sums up all options a partner has for enhancing the VJS by modifying the C/AL code for his customers.



## Use filter settings of a NAV page for the VJS

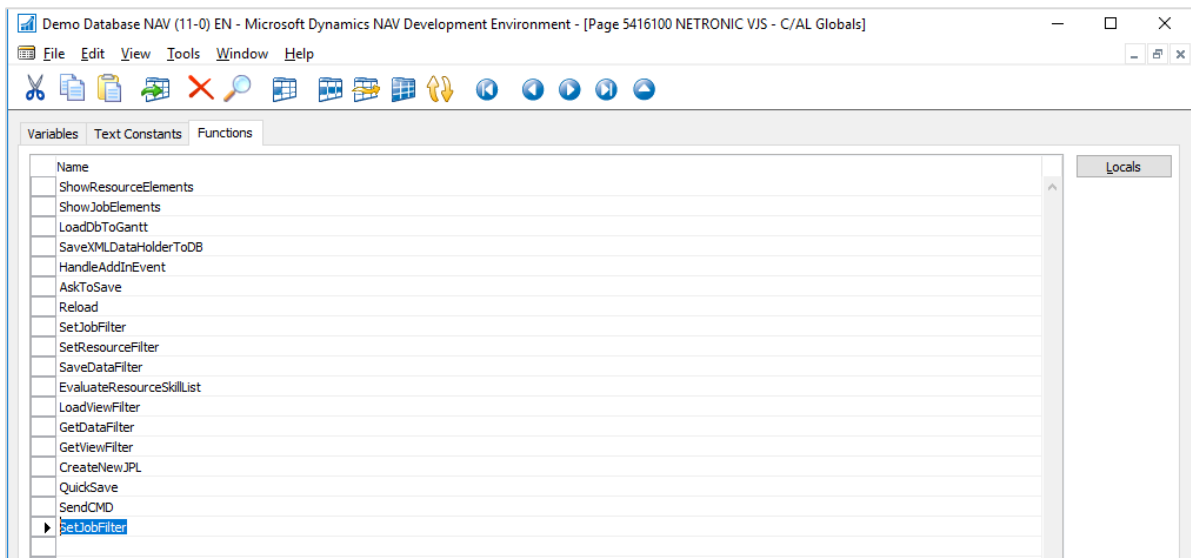
Depending on the settings, the Visual Jobs Scheduler starts with a request page where you can set up a filter for the jobs to be shown.

Proceed as described below if you want to start the VJS from a different page already containing filters or specific jobs:

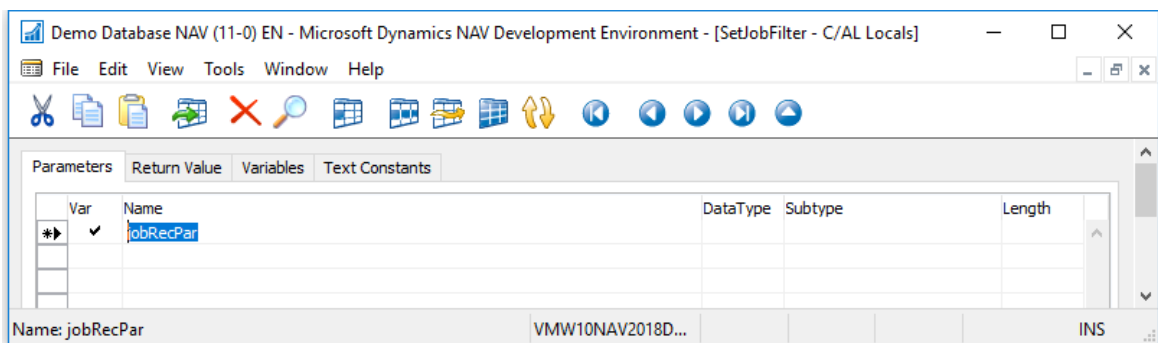
### VJS VERSION 1.1 AND BELOW

If you use Visual Jobs Scheduler **version 1.1 or below** you need to create the function on the VJS Page as well:

- Open the page “NETRONIC VJS” with Design.
- Click View → C/AL Globals.
- Open the “Functions” tab.
- Add a new function called “**SetJobFilter**” in the list.
- Please make sure that the function is **not** set to be local:  
Right-click on the line and open the properties. Set local to No.
- With the new function marked click “Locals”.



- The function needs a job Record as a Var (click into the Var column for the check mark) parameter to pass the filters over to the VJS. In this case, the parameter is called “**jobRecPar**”.



- Now close the locals and globals windows and look at the C/AL Code (F9). The body of the new VJS function should look as follows (you can copy and paste):

```
IsFilterSet := TRUE;  
JobRec.COPYFILTERS(jobRecPar);
```

- Proceed as described in Example 1 and 2.

## VJS VERSION 1.2 AND ABOVE

If you use Visual Jobs Scheduler **version 1.2 or above**, you can directly proceed as described in the following examples.

### EXAMPLE 1 ONLY SHOW PROJECT OF PROJECT CARD

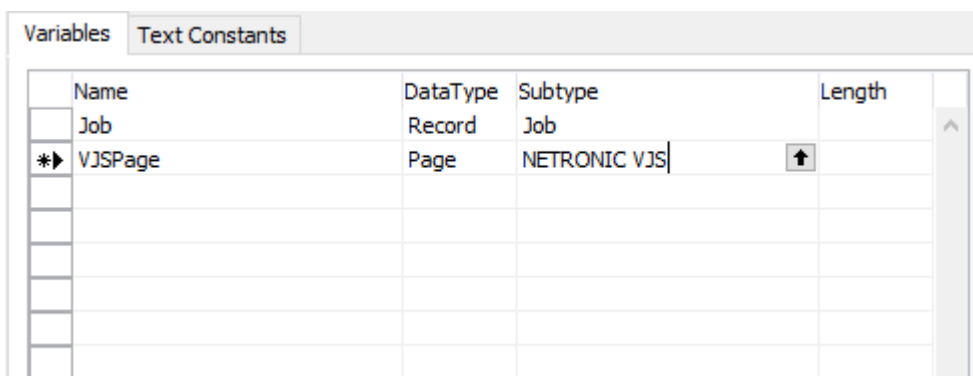
In this example we will add an action to the Page 88 “Job Card”, so that the VJS will show the job in the Job Card only.

- Design Page 88 “Job Card” and click View → “Page Actions” (Ctrl + Alt + F4).
- Now add a new action to the page.
- With the new action marked press F9 or click View → C/AL Code.

The VJS only passes filters used on the Job record so to get a job record with filters on it we need a local variable in our new action.

We also need a VJSPage variable to call our new function and to RUN.

- Click View → C/AL Locals to open the locals dialog for the action. Here you can add the two needed variables.



Variables		Text Constants	
Name	Data Type	Subtype	Length
Job	Record	Job	
*▶ VJSPage	Page	NETRONIC VJS	↑

Now for the code of the action:

```
job.SETFILTER(job."No.", "No.");  
VJSPage.SetJobFilter(job);  
VJSPage.RUN;
```

With that the action will put a filter on our local job variable, so that the “No.” field is the same. Then it passes that filter on to our new VJS function and when the page is run, the filter is used to load only that one job record.

Please note that Jobs with status “completed” will not be loaded by default!

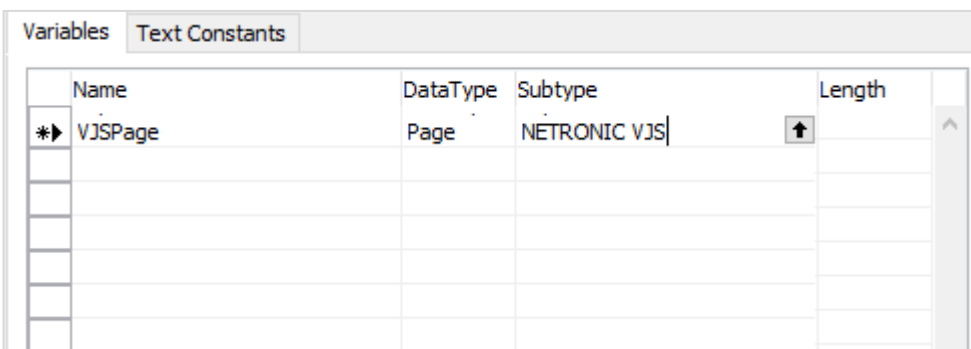
## EXAMPLE 2 – SHOW PROJECTS FROM PROJECT LIST

In this example we will add an action to the Page 89 “Job List”, so that the VJS shows the jobs that are shown in the list.

- Design Page 89 “Job List” and click View → “Page Actions” (Ctrl + Alt + F4).
- Now add a new action to the page.
- With the new action marked press F9 or click View → C/AL Code.

The VJS can use the filters used in the “Job List” Page so we only need a VJSPage variable to call our new function and to RUN.

- Click View → C/AL Locals to open the locals dialog for the action. Here you can add the needed variable.



Name	DataType	Subtype	Length
*▶ VJSPage	Page	NETRONIC VJS	1

Now for the code of the action:

```
VJSPage.SetJobFilter(Rec);  
VJSPage.RUN;
```

With those two lines the Filters that are currently used in the “Job List” Page get passed on to the VJS.

Please note that Jobs of Status “completed” will not be loaded by default, but can be activated on the Setup page!

Also please note that as of Visual Jobs Scheduler Version 1.3 you can also filter resources in the same way, with the function **SetResourceFilter**, available on the VJS Page.

## Start the VJS in the Resource View

---

The VJS starts in the Jobs View by default. If you want to change this and want to start in the Resource View instead, you only have to add one code line.

Search for the function “FillFilteredIDotNetDataHolder(...)” in codeunit 5416100 “NETRONIC VJS Management” and there look for the line:

```
DataBuilder := DataBuilder.DataBuilder;
```

Below this line add the following code:

```
DataBuilder.StartInResView := TRUE;
```



## Use date and time fields for positioning job planning lines in the diagram

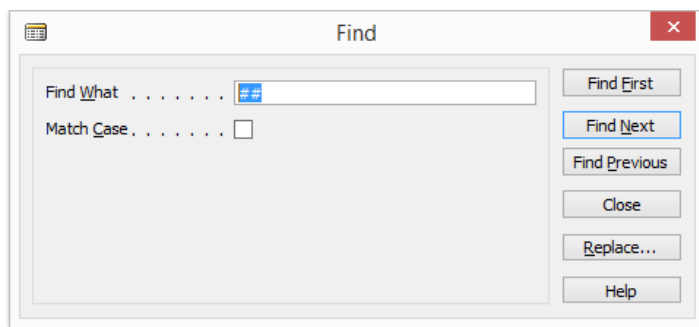
---

NAV plans job planning lines by the day. This means that they no concrete information is given as to when the job planning line should start and end at the specified date. The VJS allows to refine this planning by defining and using additional fields for date and time (start date and time, end date and time).

### STEP 1

Open codeunit 5416100 (NETRONIC VJS Management) in the Dynamics NAV Development Environment where you need to uncomment and fill out specific code lines. Since those code lines are indicated by “##”, please search for the string ## by using the built-in “Find” Function (Ctrl + F):

This code enables the VJS to load the Time information for the Job Planning Lines from the NAV table



```
//## Start and End Date and Time to show for Job Planning Line.
//## Uncomment all lines and add your Fields.
//IF (*Start Date Field Here!* <> 0D)
//AND (*End Date Field Here!* <> 0D) THEN BEGIN
//JobPlanningLine.StartDateTime :=
//CREATEDATETIME( *Start Date Field Here!* , *Start Time Field Here!*);
//JobPlanningLine.EndDateTime :=
//CREATEDATETIME( *End Date Field Here!* , *End Time Field Here!*);
//END;
//## End of code to uncomment
```

## STEP 2

Now search for the strings “##” again and modify the section shown below accordingly:

```
//## Uncomment the following line(s) and set the value DataBuilder.UseResourceCapacity
above to FALSE to use date and time in VJS.
//DataBuilder.PlanHourly := TRUE; // With this the VJS uses the start and end values and
enables the user to move bars from hour to hour.
//DataBuilder.PlanMinutewise := TRUE; // With this the user can move the bar for every
minute.
//DataBuilder.MinutesPerStep := 1; // This is the minimal amount of minutes the bar will
move in an interaction.
//DataBuilder.UseResourceTimes := TRUE; //This is to show worktime start and end as
defined below and make the bars interaction consider them
//DataBuilder.ResourceStartingTime := CREATEDATETIME(WORKDATE,080000T); //Daily
worktime start if there is capacity on that date, same for every resource
//DataBuilder.ResourceEndingTime := CREATEDATETIME(WORKDATE,180000T); //Daily
worktime end if there is capacity on that date, same for every resource
```

Setting “DataBuilder.PlanHourly := true;” enables the VJS to move Job Planning Lines at hourly intervals. If you want a more detailed control you can set the property “DataBuilder.PlanMinutewise := true;”. With this, the Job Planning Lines can be moved from minute to minute. The step size for this mode can be changed with “DataBuilder.MinutesPerStep := x” (x is Integer). If you want Job Planning Lines to move from quarter hour to quarter hour you can set this Property to 15, for instance.

## STEP 3

And now for the last section to be searched for and modified. After this, you’ll have to save your changes made in the VJS back to the NAV Table.

```
//## Start and End Date and Time to show for Job Planning Line. Uncomment needed lines and
add your fields!
//IF JobPlanningLine.CanSaveExtendedDateTimeField THEN BEGIN

//For Date and Time Fields
//VALIDATE( *Start Date Field Here!* , DT2DATE(JobPlanningLine.StartDateTime));
//VALIDATE( *End Date Field Here!* , DT2DATE(JobPlanningLine.EndDateTime));
//VALIDATE( *Start Time Field Here!* , DT2TIME(JobPlanningLine.StartDateTime));
//VALIDATE( *End Time Field Here!* , DT2TIME(JobPlanningLine.EndDateTime));

//For DateTime Fields
//VALIDATE( *Start DateTime Field Here!* , JobPlanningLine.StartDateTime);
//VALIDATE( *End DateTime Field Here!* , JobPlanningLine.EndDateTime);
// END;
//## End of Start and End Date Functionality
```

## STEP 4

Finally, you need to compile codeunit 5416100 and run the corresponding NAV Page for our Visual Jobs Scheduler

## Use milestones

---

Since NAV doesn't offer milestones neither on Job level nor on Job task level, many partners created for their customers suitable data fields in the according tables (**167** for Jobs, **1001** for Job Tasks). For visualizing these milestones also in the VJS, their usage has to be activated by modifying the data interface.

### MILESTONES ON JOB LEVEL

Milestones on Job level are used for indicating a planned delivery date (or similar) for your jobs by a grey diamond shaped symbol turning red in case the date gets exceeded. The date of the milestone symbol **can't** be modified by drag & drop

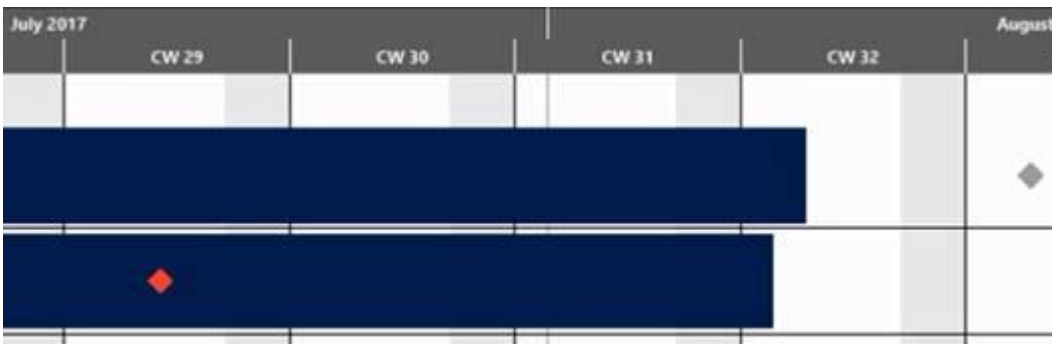
### Activate the milestones

Open codeunit 5416100 NETRONIC VJS Management where we have already prepared for you some lines of code. Supposing that you have already created the needed fields, you only have to uncomment the according code lines. For quickly finding the according section, search for the strings '++'.

```
//++ When you have a singular Milestone on the Job level
//uncomment and complete the following 2 lines
//IF JobRec."Milestone End" <> ØD THEN
// Job.ExtraMilestoneDate := CREATEDATETIME(JobRec."Milestone End", nullTime);
```

### Result

After having uncommented the code lines, a grey milestone symbol will be shown on Job level which turns red if the end date of a job planning line is later than the milestone date.



## MILESTONES ON JOB TASK LEVEL

The Job Task level milestones in the VJS are currently only representing **the start and end of a Job Task** to mark whether the Job Task is currently violating either the start or end milestone. You need not always define a pair of milestones but you could also specify only the start milestone for some Job Tasks, and both or only the end milestone for others.

### Activate the milestones

The milestones on Job Task level are also put into action in codeunit 5416100 NETRONIC VJS Management where we have already prepared for you some lines of code. Supposing that you have already created the needed fields, you only have to uncomment the according code lines. For quickly finding the according section, search for the strings '&&'.

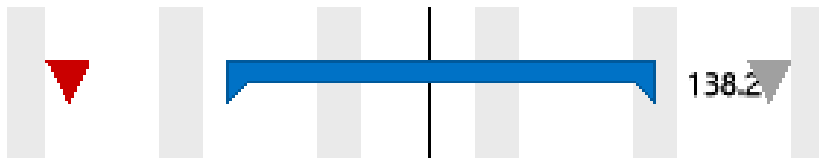
```
//&& Milestone for Job Task
//IF JobTaskRec."Milestone Start" <> 0D THEN BEGIN
//    JobTask.MileStoneStartDate := CREATEDATETIME(JobTaskRec."Milestone Start",nullTime);
//END;

//IF JobTaskRec."Milestone End" <> 0D THEN BEGIN
//    JobTask.MileStoneEndDate := CREATEDATETIME(JobTaskRec."Milestone End",nullTime);
//END;
```

### Result

After having uncommented the desired code lines, the following cases will be illustrated by milestones:

- No discrepancies (grey)
- The actual start date is later than the milestone Start (red)
- The actual end date is later than the milestone End (red)



## Handle capacities for non-project tasks

Resource capacity loads for non-project tasks, like holidays, trainings, or doctor's appointments, can be handled by a "dummy" job containing all the "other reservations" so that they will stand out from the project-related tasks (as yellow bar) in the VJS. This requires an according field in job table **167** to identify the dummy job. In our example we simply created the Boolean field "Internal Allocation".

Moreover, an explanatory text can be entered into the field "description" of the respective job planning line and will appear in the tooltip text of this "other reservations" bar.

4	26	6	0
3 3	6		

Line No. - 60000  
 Job No. - 9999  
 Planning Date - 17.01.2019  
 Type - Resource  
 No. - MARLIES  
 Description - Alloc 1  
 Quantity - 6  
 Unit of Measure Code - STUNDE  
 Job Task No. - ALLOC1  
 Quantity (Base) - 6  
 Planned Delivery Date - 17.01.2019  
 Resource Group No. -  
 Status - Order

### HOW TO PROCEED:

- You have to modify three code sections that are marked by "//--". Two of these sections can be found in codeunit **5416100 NETRONIC VJS Management**, the other one in the Page **5416102 NETRONIC VJS JPL Insert**.
- Search for these sections and uncomment them.
- Modify the sections accordingly as shown below by entering the field name declaring that this is a non-project-related job:

## Modify the sections in codeunit 5416100 NETRONIC VJS Management:

### Section 1

```
//-- Check here whether the Job is one for internal allocations uncomment and complete the
following lines
//IF JobRec.GET("Job No.") THEN BEGIN
//  IF JobRec.*Internal Allocation* THEN
//    JobPlanningLine.IsOtherAllocation := TRUE;
//END;
//-- End of code to uncomment
```

### Section 2

```
//-- Load all Jobs for internal allocations. Uncomment and complete the following lines
//JobRec.SETFILTER(JobRec.*Internal Allocation*,TRUE);
//IF JobRec.FINDSET THEN
//REPEAT
//  ProgressWindow.UPDATE(1,JobRec."No.");
//  FillDotNetJob(Job,JobRec);
//  DataBuilder.AddJob(Job);
//UNTIL JobRec.NEXT = 0;
//-- End of code to uncomment
```

## Modify the sections in Page 5416102 NETRONIC VJS JPL Insert

```
//-- Check here whether the Job is one for internal allocations uncomment and complete the
following lines
//jobRec.SETRANGE(jobRec.*Internal Allocation*, TRUE);
//
//IF jobRec.FINDSET THEN
//  REPEAT
//    IF NOT gJobRecFilter.GET(jobRec."No.") THEN BEGIN
//      gJobRecFilter.INIT;
//      gJobRecFilter.COPY(jobRec);
//      gJobRecFilter.INSERT;
//    END;
//  UNTIL jobRec.NEXT = 0;
//-- End of code to uncomment
```

## Further Resources

---

More details on the Visual Jobs Scheduler for Microsoft Dynamics NAV (2018, 2017, 2016, 2015, 2013 R2, 2013 and 2009 R2) can be found on our Website <https://www.netronic.com/dynamics-nav/visual-jobs-scheduler> or in the [partner portal](#). Feel free to contact us if you have questions, ideas or suggestions — we will be happy to answer your call or mail:

+49-(0)2408-141-0

[support@netronic.com](mailto:support@netronic.com)

## About NETRONIC - the Gantt Company

---

NETRONIC's mission is to empower every SMB organization to gain operational agility with visual scheduling. Visual scheduling is the effective management and change of time- and resource-oriented planning data and enables faster decision making, more reliable delivery time commitments, and a better use of resources. NETRONIC offers Gantt charts for visual scheduling.

As such, NETRONIC has the most complete stack of visual scheduling software.

- 1) With the **Gantt chart controls** VARCHART XGantt and VARCHART JGantt, we enable software vendors to create tailor-made Gantt charts even for complex scheduling tasks.
- 2) The **Visual Scheduling Suite** is for partners and customers of Microsoft Dynamics NAV and provides a fully integrated, intuitive visual scheduling add-ins.
- 3) ISVs developing industry solutions for Dynamics NAV and Dynamics 365 Business Central can leverage the NETRONIC NWAFF technology to integrate powerful HTML5/ JavaScript scheduler into their solutions – without writing a single line of JavaScript code.
- 4) The SaaS application **just plan it** helps SMB manufactures to achieve an easy time and resource scheduling.

For further information, please

- visit our [Website](#)
- read our [Blogs](#)
- or follow us on [Facebook](#), [Twitter](#), [Google+](#) or [LinkedIn](#) .