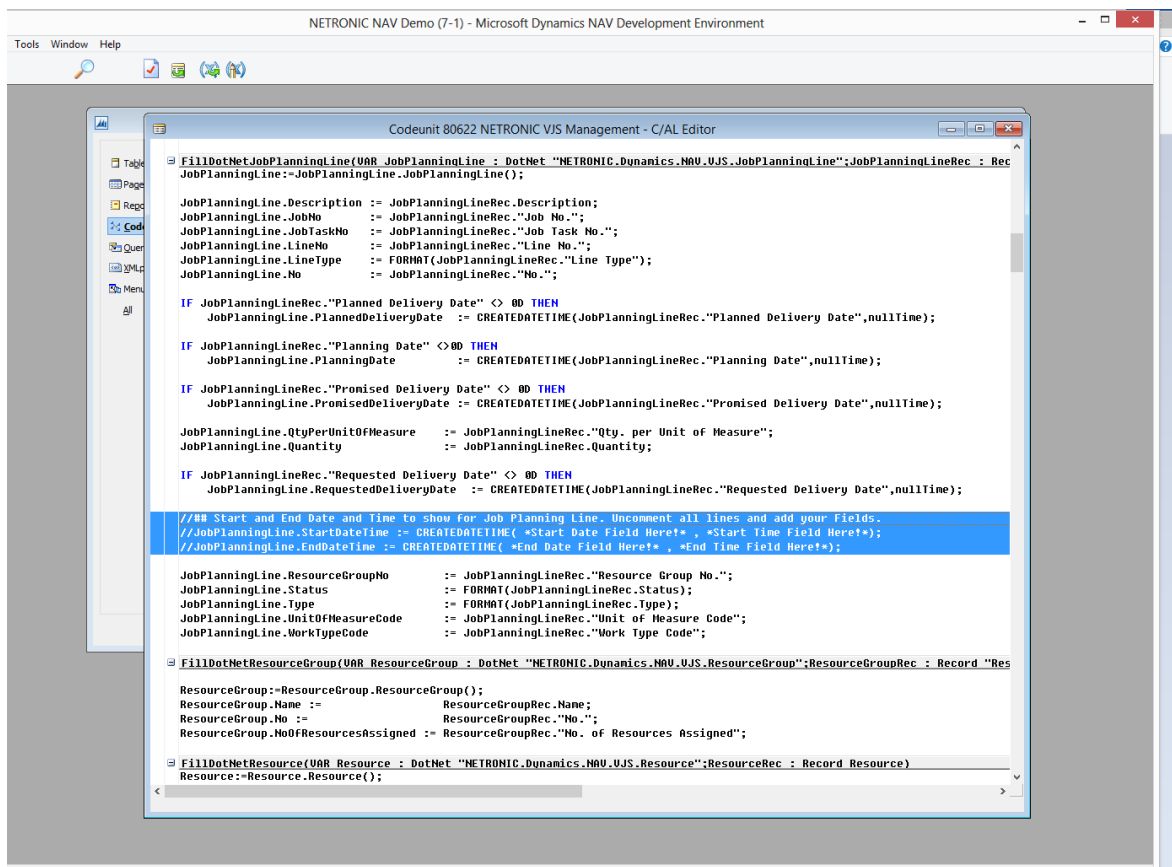


Visual Jobs Scheduler for Microsoft Dynamics NAV 2017/2016/2015/2013 R2/2013 and 2009 R2

How to ... Use other tablefields for the graphical representation in form of bars

Step 1

Open the Codeunit 80622 / 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) :



```
NETRONIC NAV Demo (7-1) - Microsoft Dynamics NAV Development Environment

Codeunit 80622 NETRONIC VJS Management - C/AL Editor

FillDotNetJobPlanningLine(VAR JobPlanningLine : DotNet "NETRONIC.Dynamics.NAV.VJS.JobPlanningLine";JobPlanningLineRec : Rec
JobPlanningLine:=JobPlanningLine.JobPlanningLine();

JobPlanningLine.Description := JobPlanningLineRec.Description;
JobPlanningLine.JobNo := JobPlanningLineRec."Job No.";
JobPlanningLine.JobTaskNo := JobPlanningLineRec."Job Task No.";
JobPlanningLine.LineNo := JobPlanningLineRec."Line No.";
JobPlanningLine.LineType := FORMAT(JobPlanningLineRec."Line Type");
JobPlanningLine.No := JobPlanningLineRec."No.";

IF JobPlanningLineRec."Planned Delivery Date" <> 00 THEN
    JobPlanningLine.PlannedDeliveryDate := CREATEDATETIME(JobPlanningLineRec."Planned Delivery Date",nullTime);

IF JobPlanningLineRec."Planning Date" <> 00 THEN
    JobPlanningLine.PlanningDate := CREATEDATETIME(JobPlanningLineRec."Planning Date",nullTime);

IF JobPlanningLineRec."Promised Delivery Date" <> 00 THEN
    JobPlanningLine.PromisedDeliveryDate := CREATEDATETIME(JobPlanningLineRec."Promised Delivery Date",nullTime);

JobPlanningLine.QtyPerUnitOfMeasure := JobPlanningLineRec."Qty. per Unit of Measure";
JobPlanningLine.Quantity := JobPlanningLineRec.Quantity;

IF JobPlanningLineRec."Requested Delivery Date" <> 00 THEN
    JobPlanningLine.RequestedDeliveryDate := CREATEDATETIME(JobPlanningLineRec."Requested Delivery Date",nullTime);

## Start and End Date and Time to show for Job Planning Line. Uncomment all lines and add your Fields.
##JobPlanningLine.StartDateAndTime := CREATEDATETIME( *Start Date Field Here* , *Start Time Field Here*);
##JobPlanningLine.EndDateAndTime := CREATEDATETIME( *End Date Field Here* , *End Time Field Here*);

JobPlanningLine.ResourceGroupNo := JobPlanningLineRec."Resource Group No.";
JobPlanningLine.Status := FORMAT(JobPlanningLineRec.Status);
JobPlanningLine.Type := FORMAT(JobPlanningLineRec.Type);
JobPlanningLine.UnitOfMeasureCode := JobPlanningLineRec."Unit of Measure Code";
JobPlanningLine.WorkTypeCode := JobPlanningLineRec."Work Type Code";

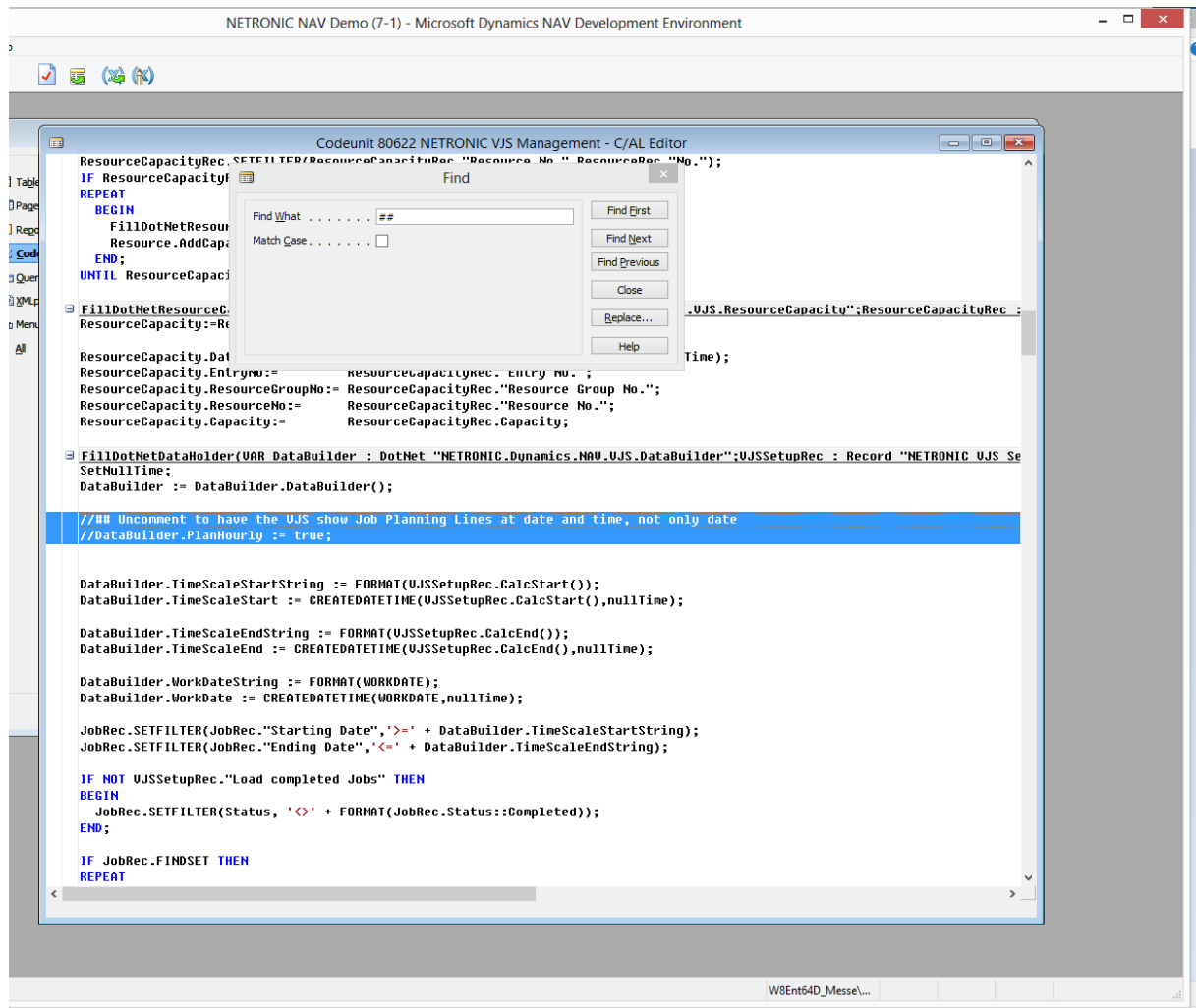
FillDotNetResourceGroup(VAR ResourceGroup : DotNet "NETRONIC.Dynamics.NAV.VJS.ResourceGroup";ResourceGroupRec : Record "Res
ResourceGroup:=ResourceGroup.ResourceGroup();
ResourceGroup.Name := ResourceGroupRec.Name;
ResourceGroup.No := ResourceGroupRec."No.";
ResourceGroup.NoOfResourcesAssigned := ResourceGroupRec."No. of Resources Assigned";

FillDotNetResource(VAR Resource : DotNet "NETRONIC.Dynamics.NAV.VJS.Resource";ResourceRec : Record Resource)
Resource:=Resource.Resource();
```

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

Step 2

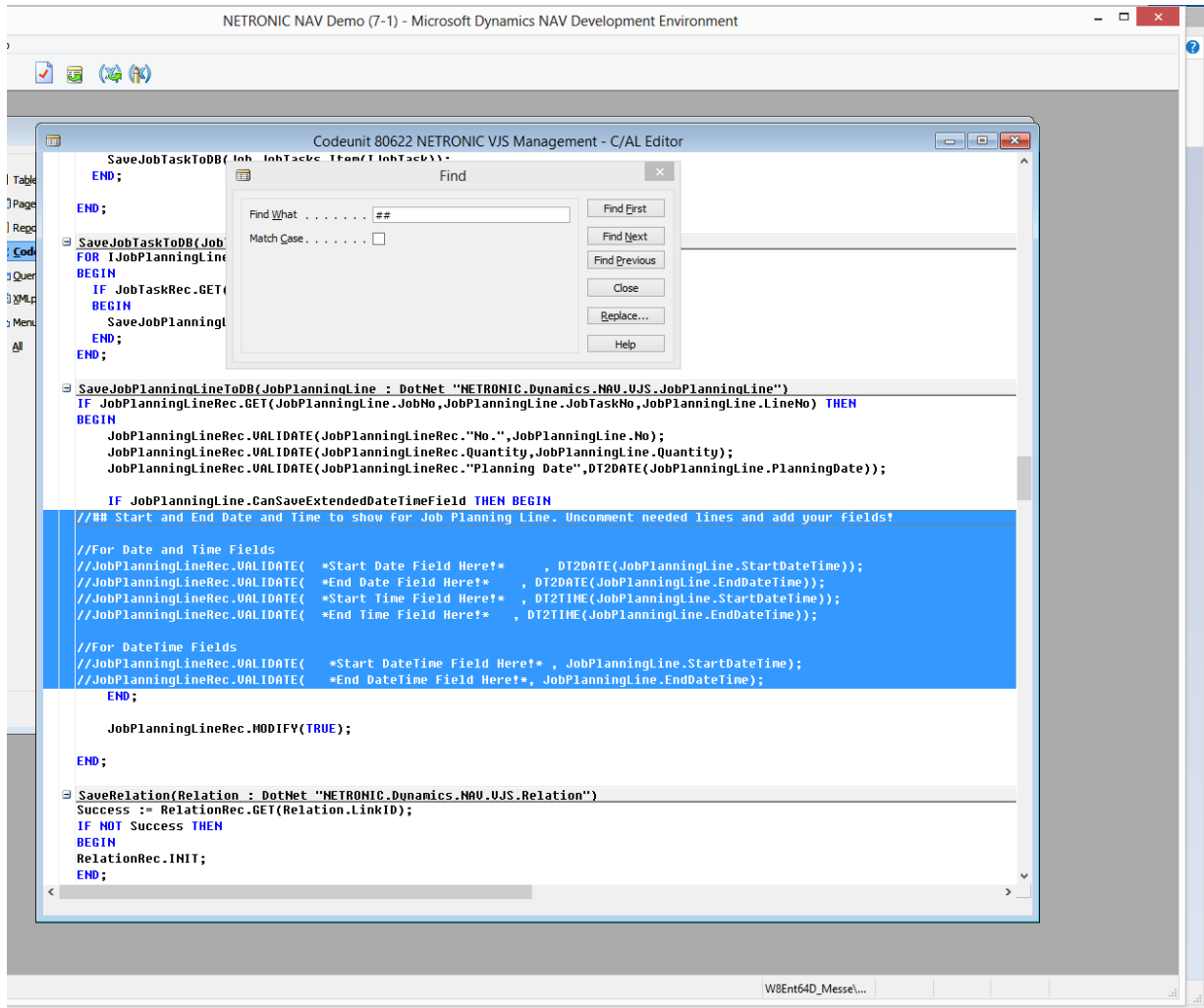
Search again for the string ##



Setting “DataBuilder.PlanHourly := true;” enables the VJS to move Job Planning Lines from hour to hour. 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.

Step 3

After the last search for ## the changes made in the VJS need to be saved back to the NAV Table.



Step 4

Finally, you need to compile the Codeunit 80622 / 5416100 and run the corresponding NAV Page for our Visual Jobs Scheduler