



Modifying Work Order Details in Maximo V7

This document details how you can modify the out of the box Work Order Details Report to meet your individual business needs. Three examples of modifications to this report will be given in the document below. These include

- A. Deleting Fields from the Planned Labor Section (Rate and Line Cost Fields)
Pages 3 - 13
- B. Deleting Fields from the Actual Labor Section (Regular Rate, Premium Rate and Line Cost Fields)
Pages 14 - 24
- C. Adding Fields to the Actual Material Section (Transaction Type Field)
Pages 25 - 34

*NOTE: Before making any modification to your report, make sure to make a backup copy of the original, out of the box report.

Additionally, this document assumes that the individual making the modifications has the applicable version of the BIRT Designer loaded and configured. For Maximo Base Services up through 7.1.1.4, BIRT Designer 2.1.2 is used. For Maximo Base Services starting with 7.1.1.5, BIRT Designer 2.3.2 is used.

Reference:

The BIRT 2.1.2 designer can be downloaded at :

https://www14.software.ibm.com/webapp/iwm/web/reg/download.do?source=tivopal&S_PKG=1TW10OT03&lang=en_US&cp=UTF-8

Information on installing and configuring the BIRT Designer for use in Maximo is located at

http://www-01.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=configuring+birt+designer+212&uid=swg21315837&loc=en_US&cs=utf-8&lang=en

The BIRT 2.3.2 designer can be downloaded at:

<http://www-01.ibm.com/software/brandcatalog/portal/opal/details?NavCode=1TW10OT07>

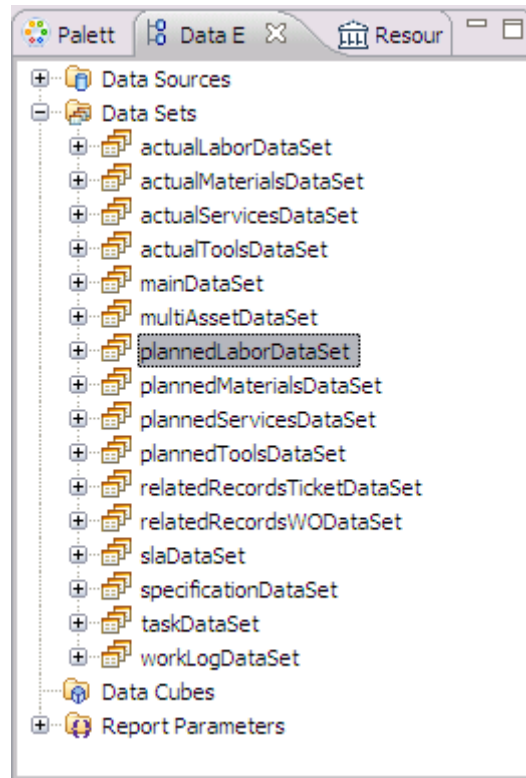
Information on installing and configuring the BIRT Designer for use in Maximo is located at

<http://www-01.ibm.com/software/sysmgmt/products/support/IBMMaximoAssetManagement.html>

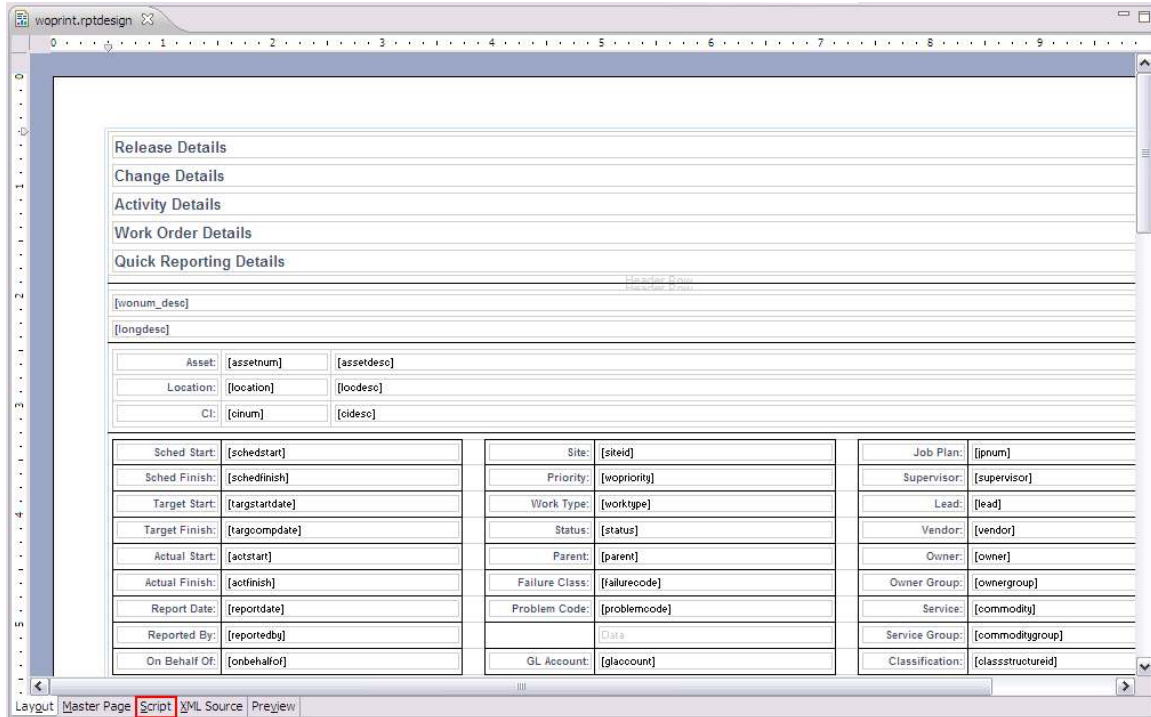
How to delete Rate and Line Cost Fields from Planned Labor table

A1. Open woprint.rpt design on BIRT

A2. On Data Explorer view, expand Data Sets and click on plannedLaborDataSet



A3. Click on Script tab




A4. Delete “wplabor.rate,” and the whole line:

“+“(wplabor.quantity * wplabor.laborhrs * wplabor.rate) as linecost, ””



A5. Switch to Fetch Script



```
Script: open
describe
fetch
close
beforeOpen

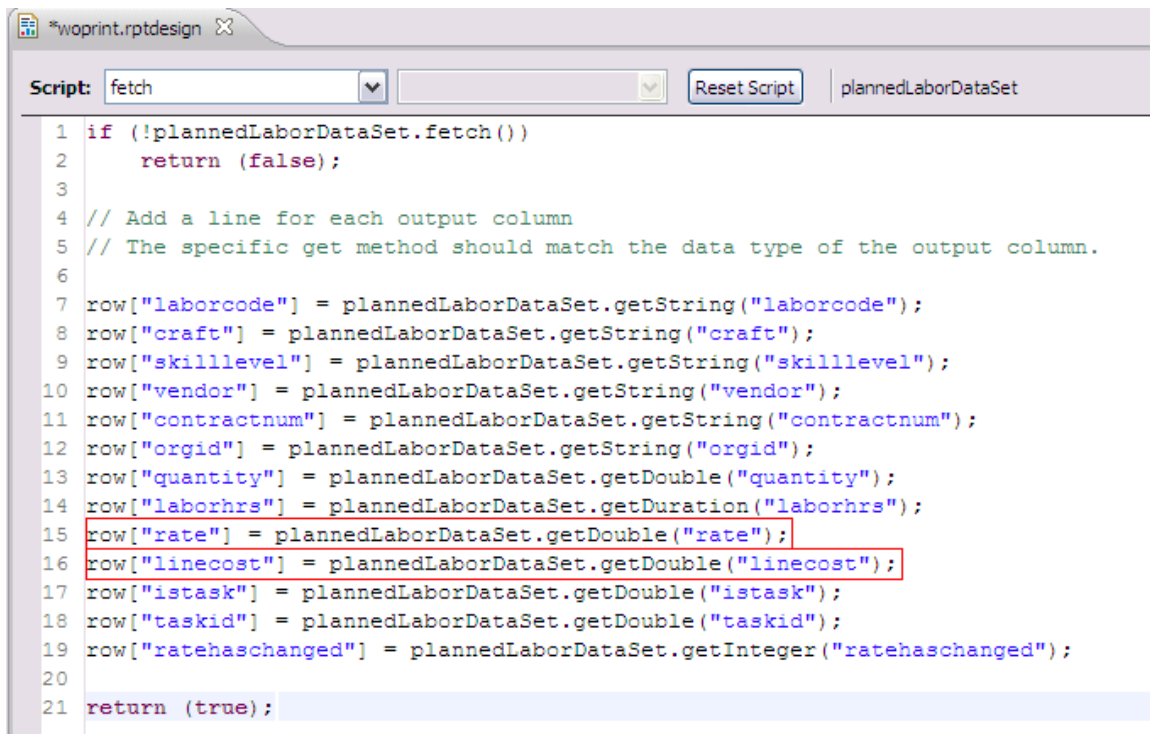
var sqlText = new String();

// Add query to sqlText variable.
sqlText = "select wplabor.laborcode,wplabor.craft, wplabor.skilllevel, wplabor.vendor, wplabor.contractnum, "
+ "wplabor.quantity, wplabor.orgid, wplabor.laborhrs, "
+ "workorder.istask, workorder.taskid, wplabor.ratehaschanged "
+ "from wplabor, workorder "
+ "where workorder.wonum = wplabor.wonum "
+ "and ((workorder.parent = ' + rows[0]["wonum"].replace('/','g','') + ' and workorder.taskid is not null) "
+ "or (workorder.wonum = ' + rows[0]["wonum"].replace('/','g','') + ' and workorder.taskid is null)) "
+ "and wplabor.siteid = workorder.siteid "
+ "and wplabor.siteid = ' + rows[0]["siteid"] + "' "
+ "and wplabor.orgid = ' + rows[0]["orgid"] + "' "
+ "order by workorder.taskid "
;

plannedLaborDataSet.setQuery(sqlText);
```

A6. Delete lines:

```
row["rate"] = plannedLaborDataSet.getDouble("rate");
row["linecost"] = plannedLaborDataSet.getDouble("linecost");
```

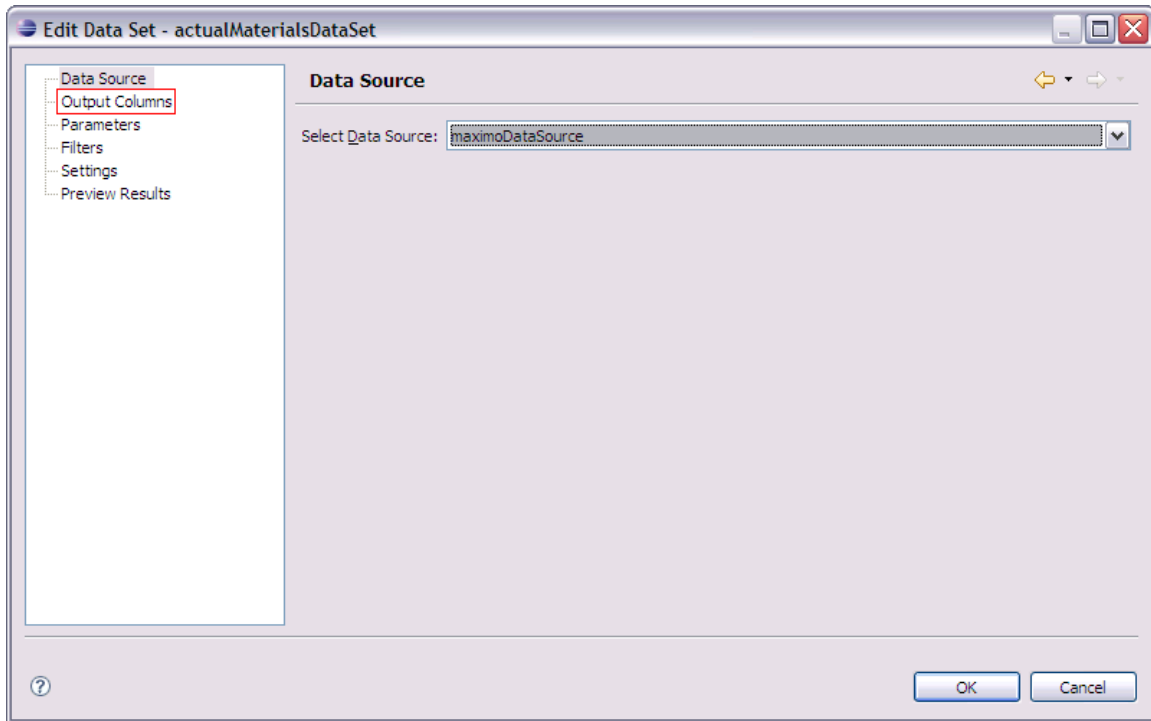


```
Script: fetch

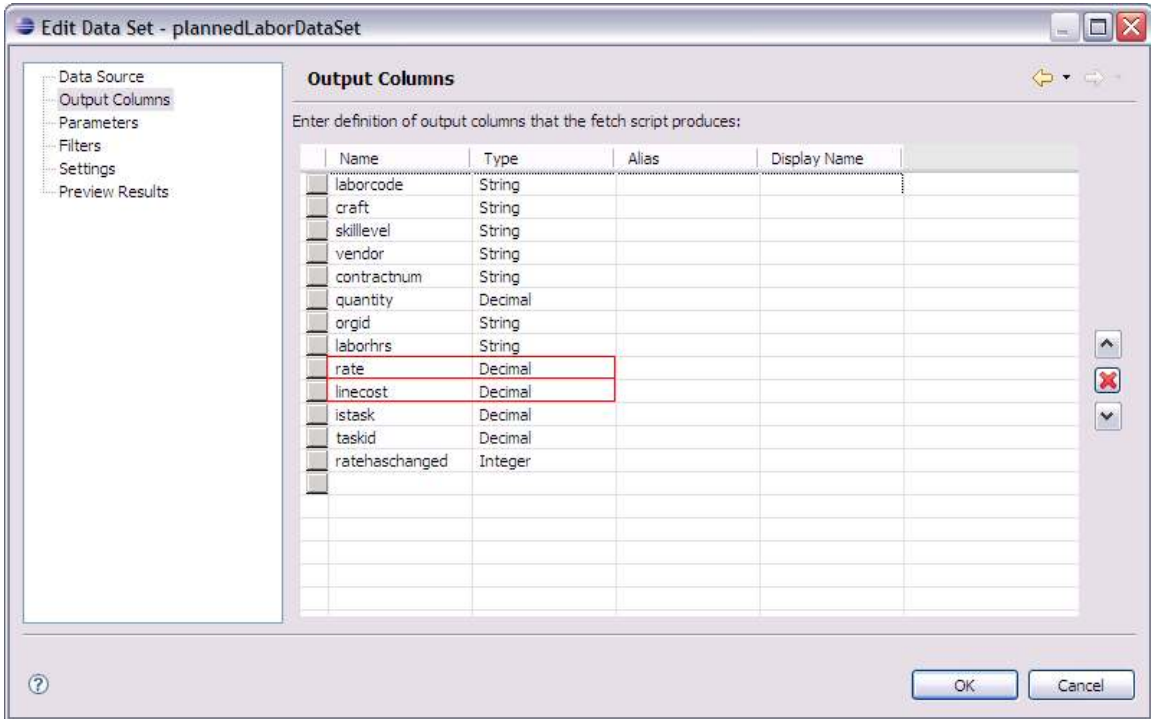
1 if (!plannedLaborDataSet.fetch())
2     return (false);
3
4 // Add a line for each output column
5 // The specific get method should match the data type of the output column.
6
7 row["laborcode"] = plannedLaborDataSet.getString("laborcode");
8 row["craft"] = plannedLaborDataSet.getString("craft");
9 row["skilllevel"] = plannedLaborDataSet.getString("skilllevel");
10 row["vendor"] = plannedLaborDataSet.getString("vendor");
11 row["contractnum"] = plannedLaborDataSet.getString("contractnum");
12 row["orgid"] = plannedLaborDataSet.getString("orgid");
13 row["quantity"] = plannedLaborDataSet.getDouble("quantity");
14 row["laborhrs"] = plannedLaborDataSet.getDuration("laborhrs");
15 row["rate"] = plannedLaborDataSet.getDouble("rate");
16 row["linecost"] = plannedLaborDataSet.getDouble("linecost");
17 row["istask"] = plannedLaborDataSet.getDouble("istask");
18 row["taskid"] = plannedLaborDataSet.getDouble("taskid");
19 row["ratehaschanged"] = plannedLaborDataSet.getInteger("ratehaschanged");
20
21 return (true);
```

A7. On Data Explorer, double click plannedLaborDataSet Data Set

A8. Click on Output Columns



A9. Delete both rate and linecost entries by selecting each one and clicking on “X” button. Click on OK button



A10. Click on Layout tab

```

1  if (!plannedLaborDataSet.fetch())
2      return (false);
3
4  // Add a line for each output column
5  // The specific get method should match the data type of the output column.
6
7  row["laborcode"] = plannedLaborDataSet.getString("laborcode");
8  row["craft"] = plannedLaborDataSet.getString("craft");
9  row["skilllevel"] = plannedLaborDataSet.getString("skilllevel");
10 row["vendor"] = plannedLaborDataSet.getString("vendor");
11 row["contractnum"] = plannedLaborDataSet.getString("contractnum");
12 row["orgid"] = plannedLaborDataSet.getString("orgid");
13 row["quantity"] = plannedLaborDataSet.getDouble("quantity");
14 row["laborhrs"] = plannedLaborDataSet.getDuration("laborhrs");
15 row["rate"] = plannedLaborDataSet.getDouble("rate");
16 row["linecost"] = plannedLaborDataSet.getDouble("linecost");
17 row["istask"] = plannedLaborDataSet.getDouble("istask");
18 row["taskid"] = plannedLaborDataSet.getDouble("taskid");
19 row["ratehaschanged"] = plannedLaborDataSet.getInteger("ratehaschanged");
20
21 return (true);

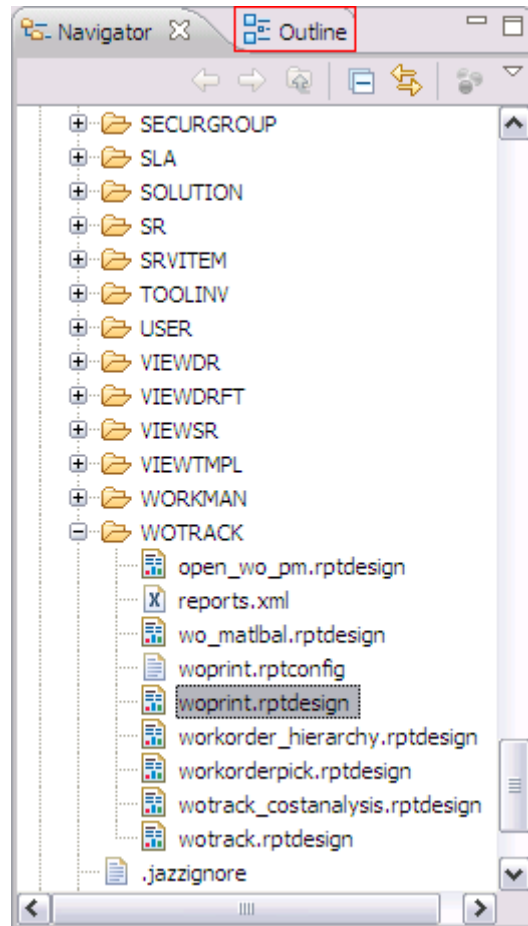
```

A11. On the report layout, click on Planned Labor label field

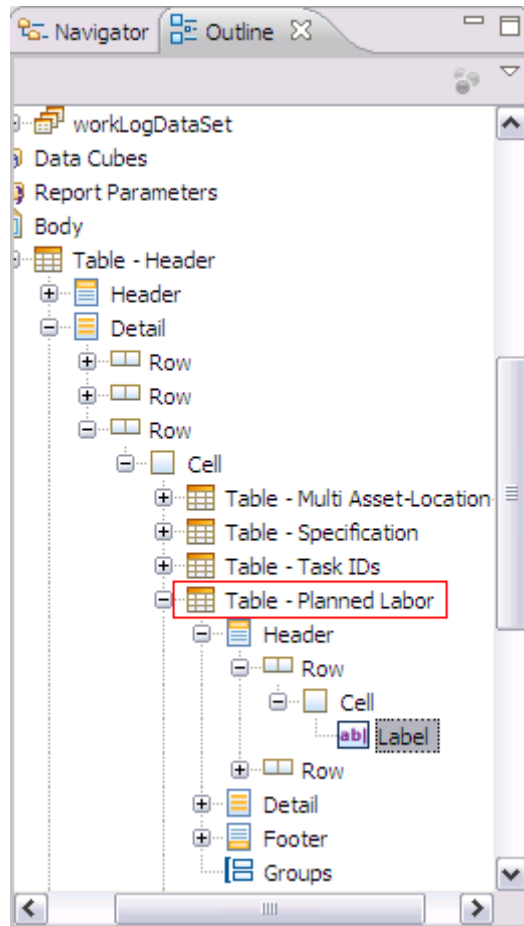
Planned Labor										
Task ID	Craft	Skill Level	Labor	Vendor	Contract	Qty	Hours	Rate	Line Cost	
[taskid]	[craft]	[skilllevel]	[laborcode]	[vendor]	[contractnum]	[quantity]	[laborhrs]	[rate]	[linecost]	
Footer Row									Total Planned Labor:	[data item]

Planned Materials						
Task ID	Item	Description	Storeroom	Qty	Unit Cost	Line Cost
[taskid]	[itemnum]	[description]	[location]	[itemqty]	[unitcost]	[linecost]
Footer Row						Total Planned Materials: [data item]

A12. Click on Outline tab



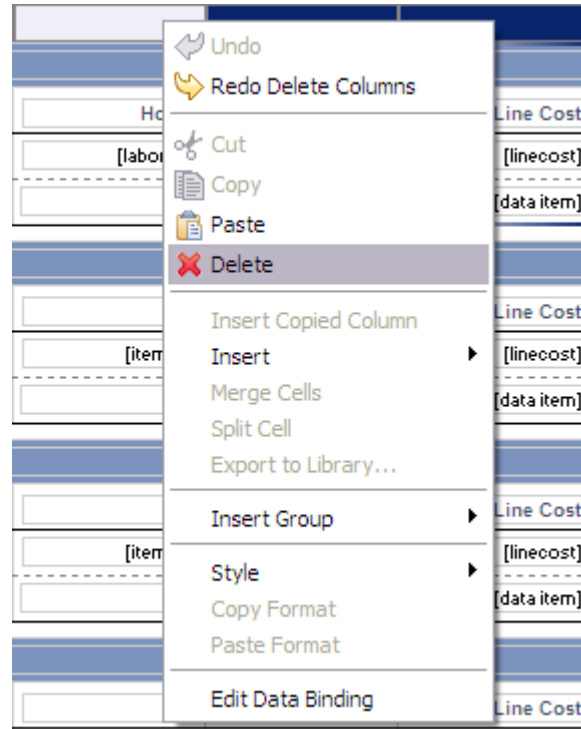
A13. Select Table – Planned Labor



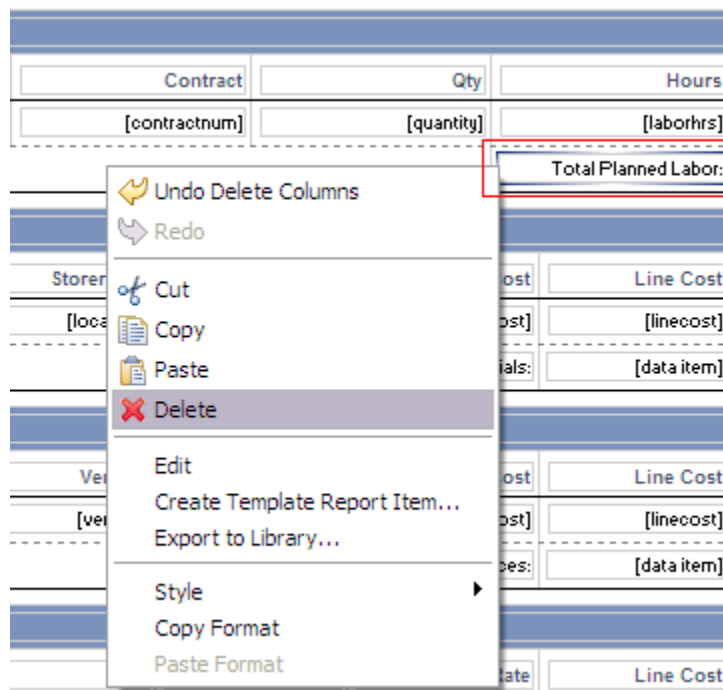
A14. Select the top of both Rate and Line Cost columns

[measurementvalue]		[measuredate]	[observation]
Qty	Hours	Rate	Line Cost
[quantity]	[laborhrs]	[rate]	[linecost]
		Total Planned Labor:	[data item]


A15. Right click on it and select Delete

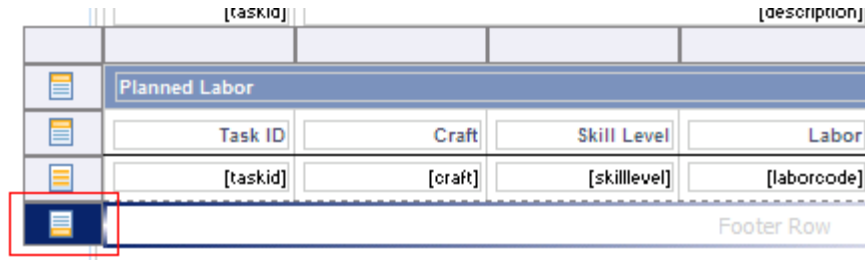


A16. Right click on “Total Planned Labor:” label and select Delete



A17. Select the Planned table again (steps 11, 12 and 13)

A18. On Planned table, select the icon  from the last row (Footer Row)



[taskid]	[description]		
Planned Labor			
Task ID	Craft	Skill Level	Labor
[taskid]	[craft]	[skilllevel]	[laborcode]
Footer Row			

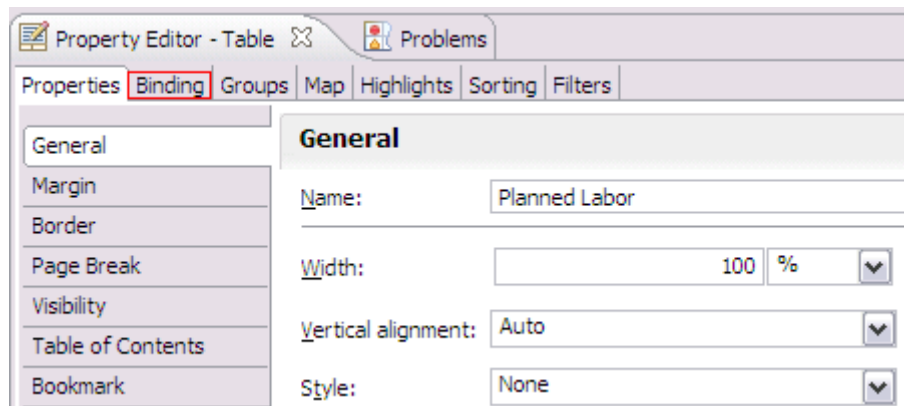
A19. On Property Editor view, General tab, set the Height to 0.08

A20. The table should look like this:



Task ID	Craft	Skill Level	Labor	Vendor	Contract	Qty	Hours
[taskid]	[craft]	[skilllevel]	[laborcode]	[vendor]	[contractnum]	[quantity]	[laborhrs]

A21. On Property Editor, with the table selected, click on Binding tab



Property Editor - Table Problems

Properties **Binding** Groups Map Highlights Sorting Filters

General

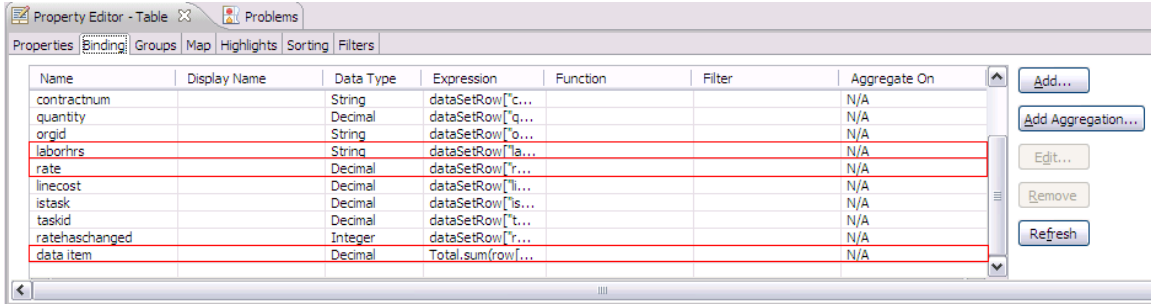
Name: Planned Labor

Width: 100 %

Vertical alignment: Auto

Style: None

A22. Delete rate, linecost and data item rows

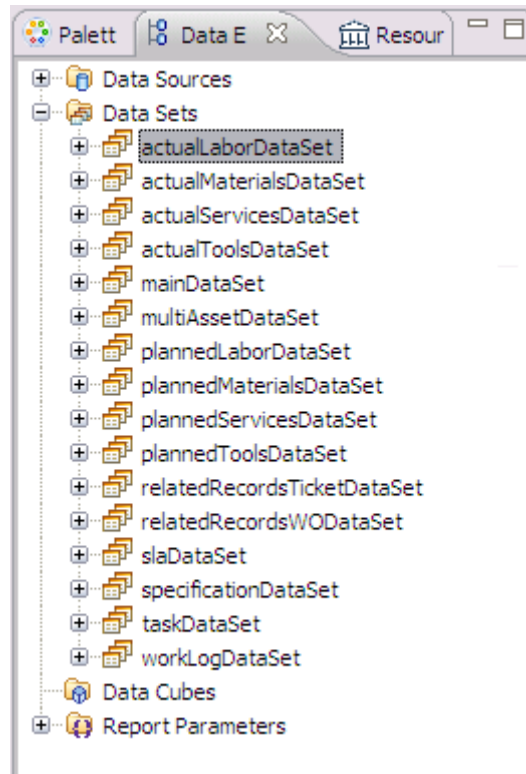


A23. Save the report.

How to delete Regular Rate, Premium Rate and Line Cost Fields from Actual Labor table

B1. Open woprint.rpt design on BIRT

B2. On Data Explorer view, expand Data Sets and click on actualLaborDataSet



B3. Click on Script tab

The screenshot shows the 'woprnt.rptdesign' application window. The 'Script' tab is highlighted in the bottom navigation bar. The main content area contains a form with the following sections:

- Release Details
- Change Details
- Activity Details
- Work Order Details
- Quick Reporting Details

Fields include: [wonom_desc], [longdesc], Asset: [assetnum] [assetdesc], Location: [location] [locdesc], CI: [cinum] [cidesc], Sched Start: [schedstart], Site: [siteid], Job Plan: [jplanum], Sched Finish: [schedfinish], Priority: [wopriority], Supervisor: [supervisor], Target Start: [targetstart], Work Type: [worktype], Lead: [lead], Target Finish: [targetfinish], Status: [status], Vendor: [vendor], Actual Start: [actstart], Parent: [parent], Owner: [owner], Actual Finish: [actfinish], Failure Class: [failurecode], Owner Group: [ownergroup], Report Date: [reportdate], Problem Code: [problecode], Service: [commodity], Reported By: [reportedby], GL Account: [glaccount], Service Group: [commoditygroup], On Behalf Of: [onbehalfof], Classification: [classstructureid].

B4. Delete “labtrans.payrate,” and the whole line:

“+ "labtrans.premiumpayratetype, labtrans.linecost, labtrans.premiumpayrate ”

The screenshot shows the 'woprnt.rptdesign' application window with the 'Script' tab selected. The script editor displays the following code:

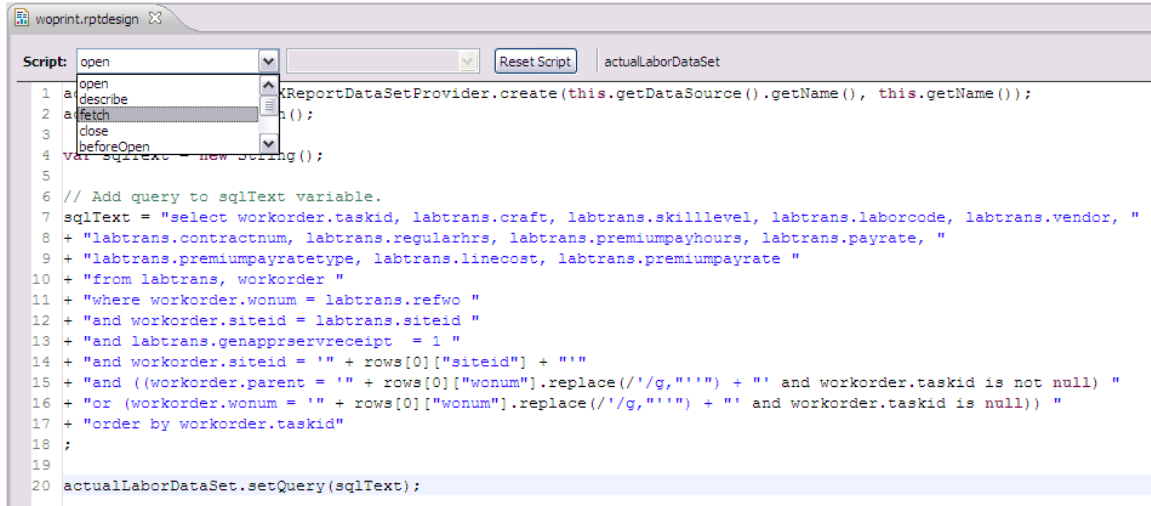
```

1 actualLaborDataSet = MXReportDataSetProvider.create(this.getDataSource().getName(), this.getName());
2 actualLaborDataSet.open();
3
4 var sqlText = new String();
5
6 // Add query to sqlText variable.
7 sqlText = "select workorder.taskid, labtrans.craft, labtrans.skilllevel, labtrans.laborcode, labtrans.vendor, "
8 + "labtrans.contractnum, labtrans.regularhrs, labtrans.premiumpayhours, labtrans.payrate, "
9 + "labtrans.premiumpayratetype, labtrans.linecost, labtrans.premiumpayrate "
10 + "from labtrans, workorder "
11 + "where workorder.wonum = labtrans.refwo "
12 + "and workorder.siteid = labtrans.siteid "
13 + "and labtrans.genapprservreceipt = 1 "
14 + "and workorder.siteid = '" + rows[0]["siteid"] + "'"
15 + "and ((workorder.parent = '" + rows[0]["wonum"].replace('/',g,'') + "' and workorder.taskid is not null) "
16 + "or (workorder.wonum = '" + rows[0]["wonum"].replace('/',g,'') + "' and workorder.taskid is null)) "
17 + "order by workorder.taskid"
18 ;
19
20 actualLaborDataSet.setQuery(sqlText);

```

The line 9 is highlighted in red in the original image, indicating the deletion of 'labtrans.payrate' and the addition of 'labtrans.premiumpayratetype, labtrans.linecost, labtrans.premiumpayrate'.

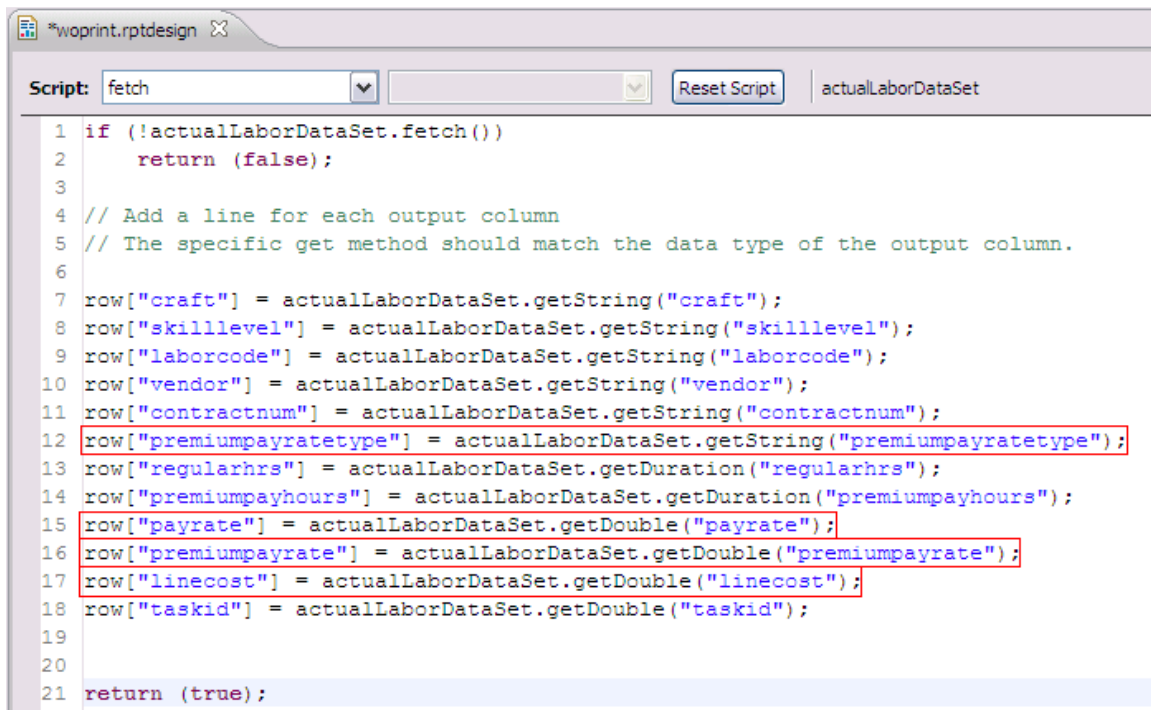
B5. Switch to Fetch Script



```
Script: open [dropdown] [dropdown] [Reset Script] actualLaborDataSet
1 a
2 describe
3 a fetch
4 close
5 beforeOpen
6 var sqlText = new String();
7
8 // Add query to sqlText variable.
9 sqlText = "select workorder.taskid, labtrans.craft, labtrans.skilllevel, labtrans.laborcode, labtrans.vendor, "
10 + "labtrans.contractnum, labtrans.regularhrs, labtrans.premiumpayhours, labtrans.payrate, "
11 + "labtrans.premiumpayratetype, labtrans.linecost, labtrans.premiumpayrate "
12 + "from labtrans, workorder "
13 + "where workorder.wonum = labtrans.refwo "
14 + "and workorder.siteid = labtrans.siteid "
15 + "and labtrans.genapprservreceipt = 1 "
16 + "and workorder.siteid = '" + rows[0]["siteid"] + "' "
17 + "and ((workorder.parent = '" + rows[0]["wonum"].replace('/g','') + "' and workorder.taskid is not null) "
18 + "or (workorder.wonum = '" + rows[0]["wonum"].replace('/g','') + "' and workorder.taskid is null)) "
19 + "order by workorder.taskid"
20 ;
21 actualLaborDataSet.setQuery(sqlText);
```

B6. Delete lines:

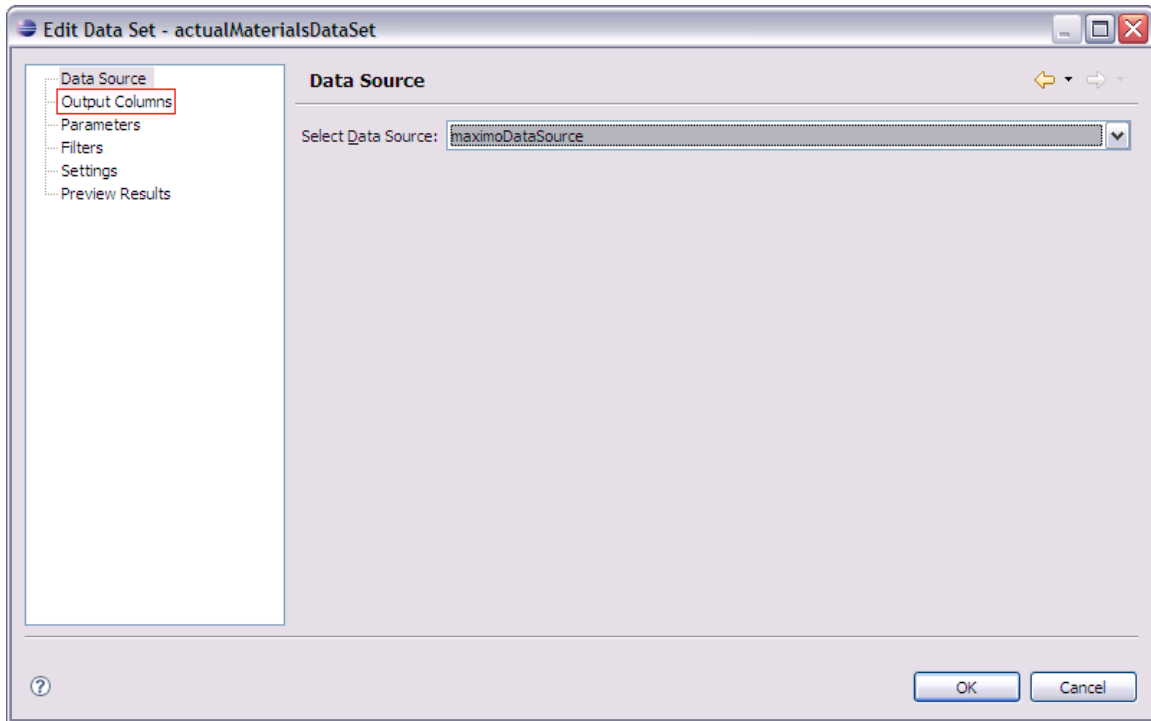
```
row["premiumpayratetype"] = actualLaborDataSet.getString("premiumpayratetype");
row["payrate"] = actualLaborDataSet.getDouble("payrate");
row["premiumpayrate"] = actualLaborDataSet.getDouble("premiumpayrate");
row["linecost"] = actualLaborDataSet.getDouble("linecost");
```



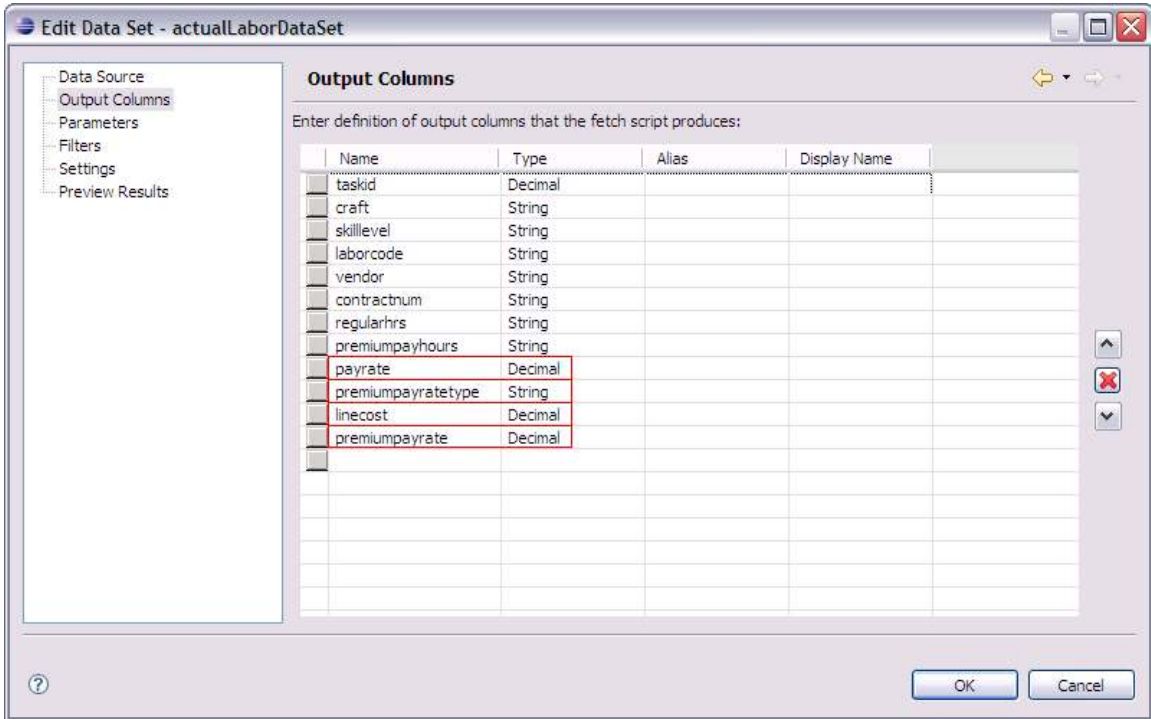
```
*woprint.rptdesign [dropdown] [dropdown] [Reset Script] actualLaborDataSet
Script: fetch
1 if (!actualLaborDataSet.fetch())
2     return (false);
3
4 // Add a line for each output column
5 // The specific get method should match the data type of the output column.
6
7 row["craft"] = actualLaborDataSet.getString("craft");
8 row["skilllevel"] = actualLaborDataSet.getString("skilllevel");
9 row["laborcode"] = actualLaborDataSet.getString("laborcode");
10 row["vendor"] = actualLaborDataSet.getString("vendor");
11 row["contractnum"] = actualLaborDataSet.getString("contractnum");
12 row["premiumpayratetype"] = actualLaborDataSet.getString("premiumpayratetype");
13 row["regularhrs"] = actualLaborDataSet.getDuration("regularhrs");
14 row["premiumpayhours"] = actualLaborDataSet.getDuration("premiumpayhours");
15 row["payrate"] = actualLaborDataSet.getDouble("payrate");
16 row["premiumpayrate"] = actualLaborDataSet.getDouble("premiumpayrate");
17 row["linecost"] = actualLaborDataSet.getDouble("linecost");
18 row["taskid"] = actualLaborDataSet.getDouble("taskid");
19
20
21 return (true);
```


B7. On Data Explorer, double click actualLaborDataSet Data Set

B8. Click on Output Columns



B9. Delete payrate, premiumpayratetype, linecost and premiumpayrate entries by selecting each one and clicking on “X” button. Click on OK button



B10. Click on Layout tab

The screenshot shows the 'woprnt.rptdesign' application window. At the top, there is a 'Script:' dropdown menu set to 'fetch', a 'Reset Script' button, and the text 'actualLaborDataSet'. Below this is a code editor with the following script:

```

1  if (!actualLaborDataSet.fetch())
2      return (false);
3
4  // Add a line for each output column
5  // The specific get method should match the data type of the output column.
6
7  row["craft"] = actualLaborDataSet.getString("craft");
8  row["skilllevel"] = actualLaborDataSet.getString("skilllevel");
9  row["laborcode"] = actualLaborDataSet.getString("laborcode");
10 row["vendor"] = actualLaborDataSet.getString("vendor");
11 row["contractnum"] = actualLaborDataSet.getString("contractnum");
12 row["regularhrs"] = actualLaborDataSet.getDuration("regularhrs");
13 row["premiumpayhours"] = actualLaborDataSet.getDuration("premiumpayhours");
14 row["taskid"] = actualLaborDataSet.getDouble("taskid");
15
16
17 return (true);

```

At the bottom of the editor, there are five tabs: 'Layout' (highlighted with a red box), 'Master Page', 'Script', 'XML Source', and 'Preview'.

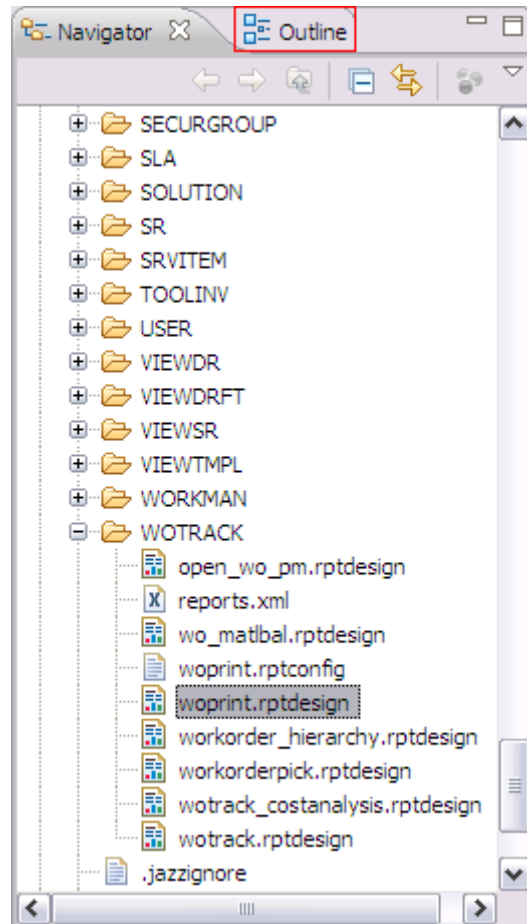
B11. On the report layout, click on Actual Labor label field

The screenshot shows the report layout in the 'woprnt.rptdesign' application. The layout is divided into several sections:

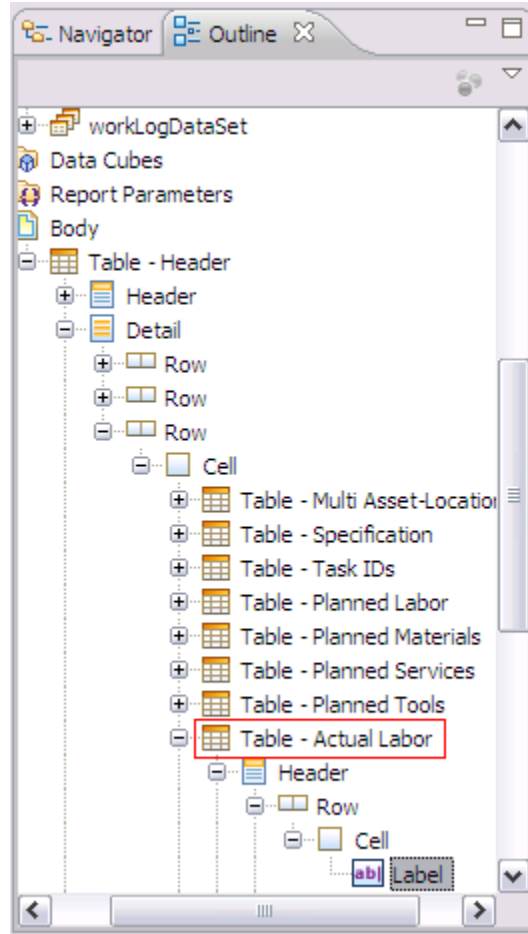
- Planned Tools:** A table with columns: Task ID, Tool, Description, Qty, Hrs, Rate, Line Cost. Below the table is a 'Footer Row' with 'Total Planned Tools: [data item]'.
- Actual Labor:** A table with columns: Task ID, Craft, Skill Level, Labor, Vendor, Contract Num, Regular, Premium Hours, Regular, Premium Rate, Line Cost. Below the table is a 'Footer Row' with 'Total Actual Labor: [data item]'.
- Actual Materials:** A table with columns: Task ID, Item, Description, Storeroom, Qty, Unit Cost, Line Cost, Issue Type. Below the table is a 'Footer Row' with 'Total Actual Materials: [data item]'.

The 'Actual Labor' section is highlighted with a red border, and the 'Actual Labor' label field is the target of the instruction.

B12. Click on Outline tab



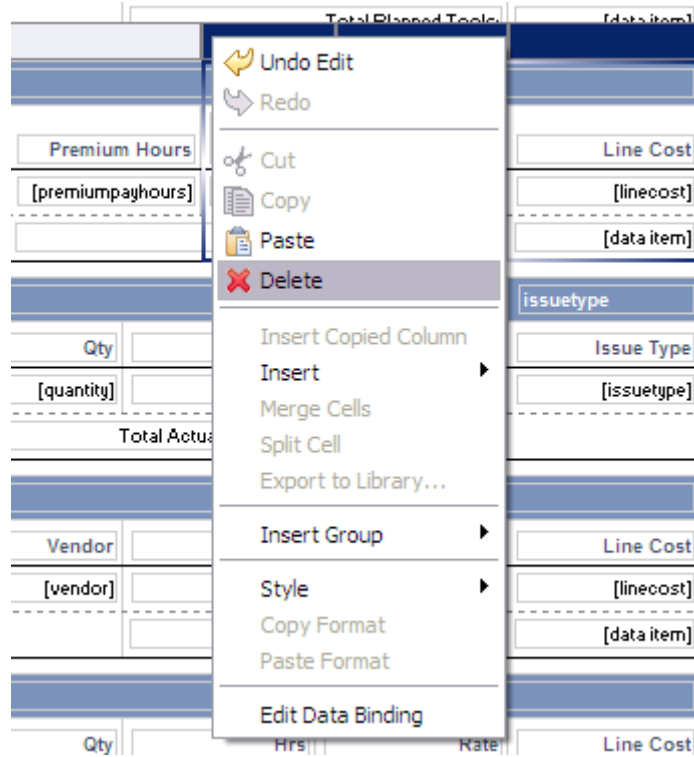
B13. Select Table – Actual Labor



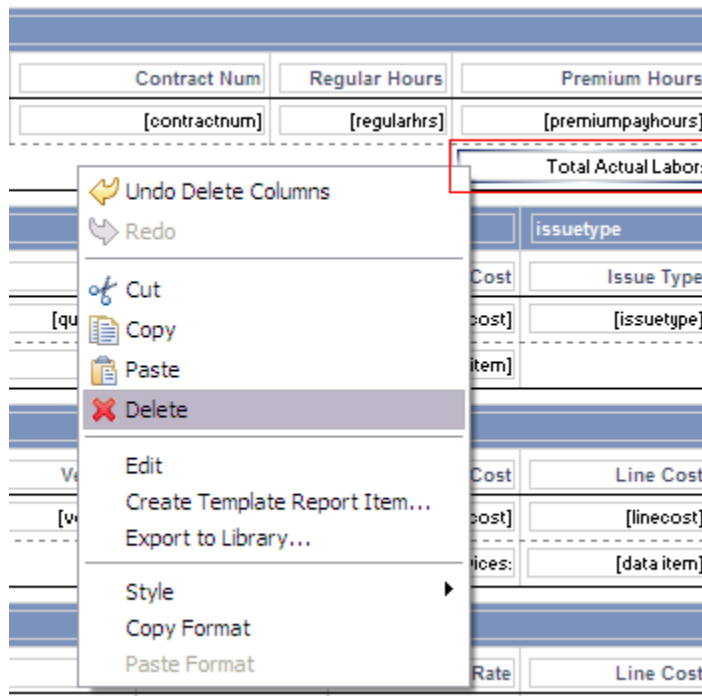
B14. Select the top of Regular, Premium Rate and Line Cost columns

		Total Planned Tools:		[data item]
Regular	Premium Hours	Regular	Premium Rate	Line Cost
[regularhrs]	[premiumpayhours]	[payrate]	[premiumpayrate]	[linecost]
Total Actual Labor:				[data item]

B15. Right click on it and select Delete







B16. Right click on “Total Actual Labor:” label and select Delete



B17. Select the Actual table again (steps 11, 12 and 13)

B18. On Actual Labor table, select the icon  from the last row (Footer Row)

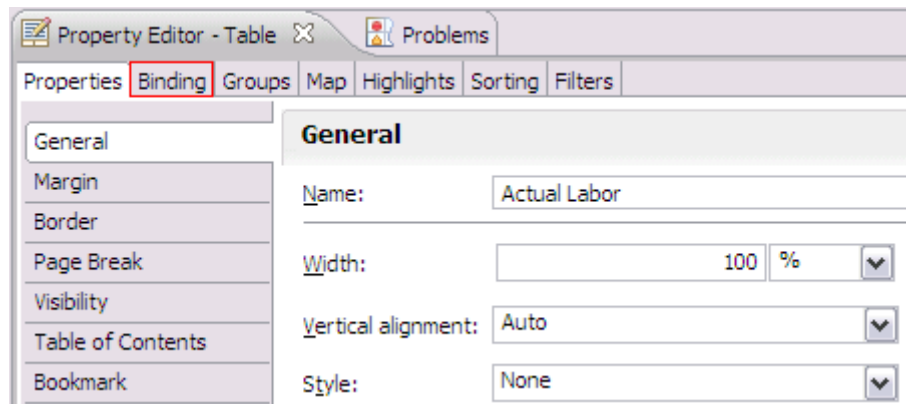
	[taskid]			[description]
	Planned Labor			
	Task ID	Craft	Skill Level	Labor
	[taskid]	[craft]	[skilllevel]	[laborcode]
	Footer Row			

B19. On Property Editor view, General tab, set the Height to 0.08

B20. The table should look like this:

Actual Labor								
Task ID	Craft	Skill Level	Labor	Vendor	Contract Num	Regular Hours	Premium Hours	
[taskid]	[craft]	[skilllevel]	[laborcode]	[vendor]	[contractnum]	[regularhrs]	[premiumpghours]	

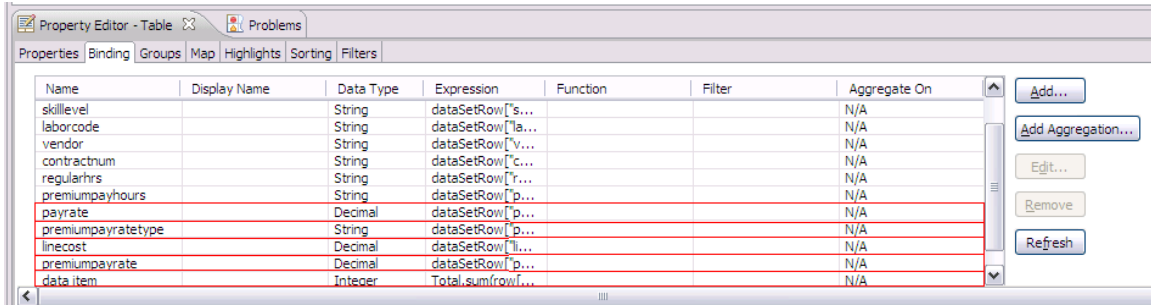
B21. On Property Editor, with the table selected, click on Binding tab



The screenshot shows the 'Property Editor - Table' window with the 'Binding' tab selected. The 'General' section is visible, showing the following settings:

- Name: Actual Labor
- Width: 100 %
- Vertical alignment: Auto
- Style: None

B22. Delete rate, linecost and data item rows

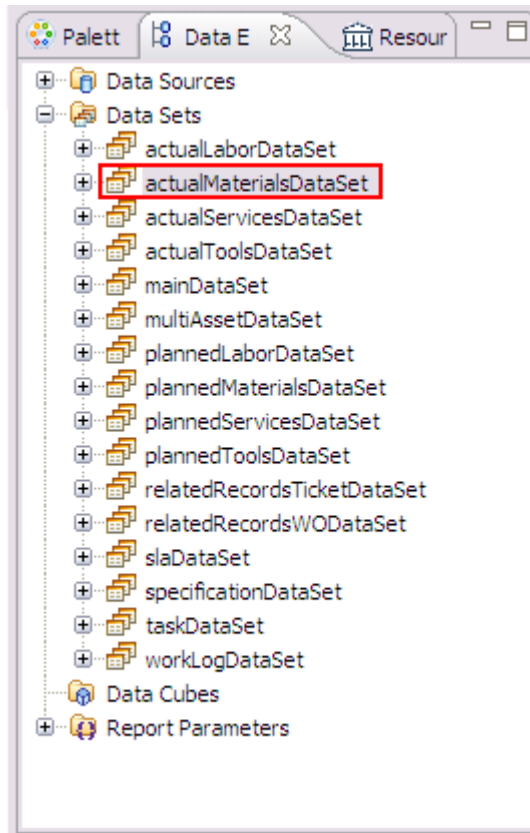


B23. Save the report.

How to add Transaction Type field on Actual Materials Table

C1. Open woprint.rpt desing on BIRT.

C2. On Data Explorer view, expand Data Sets and click on actualMaterialsDataSet.



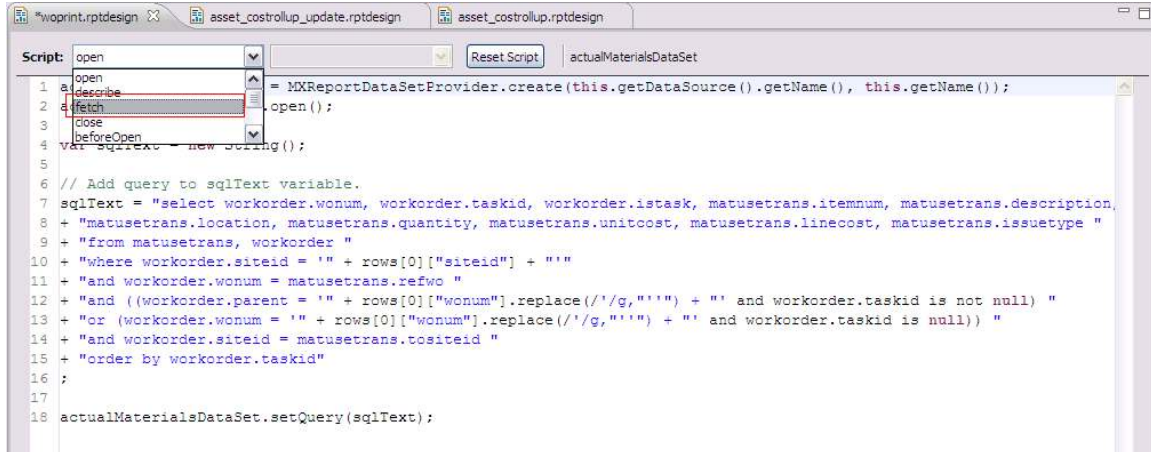
C3. Click on Script tab

The screenshot shows the 'woprnt.rptdesign' application window. The main area contains a form with several sections: Release Details, Change Details, Activity Details, Work Order Details, and Quick Reporting Details. Below these sections are various input fields for data like [wonom_desc], [longdesc], Asset, Location, CI, Sched Start, Site, Job Plan, etc. At the bottom, a navigation bar includes 'Layout', 'Master Page', 'Script' (highlighted with a red box), 'XML Source', and 'Preview'.

C4. Add “, matusetrans.issueType” before the “from” clause.

```
Script: open [dropdown] [Reset Script] actualMaterialsDataSet
1 actualMaterialsDataSet = MXReportDataSetProvider.create(this.getDataSource().getName(), this.getName());
2 actualMaterialsDataSet.open();
3
4 var sqlText = new String();
5
6 // Add query to sqlText variable.
7 sqlText = "select workorder.wonum, workorder.taskid, workorder.istask, matusetrans.itemnum, matusetrans.description,
8 + "matusetrans.location, matusetrans.quantity, matusetrans.unitcost, matusetrans.linecost, matusetrans.issueType "
9 + "from matusetrans, workorder "
10 + "where workorder.siteid = '" + rows[0]["siteid"] + "'
11 + "and workorder.wonum = matusetrans.refwo "
12 + "and ((workorder.parent = '" + rows[0]["wonum"].replace('/',g,'') + "' and workorder.taskid is not null) "
13 + "or (workorder.wonum = '" + rows[0]["wonum"].replace('/',g,'') + "' and workorder.taskid is null) "
14 + "and workorder.siteid = matusetrans.tositeid "
15 + "order by workorder.taskid"
16 ;
17
18 actualMaterialsDataSet.setQuery(sqlText);
```


C5. Switch to Fetch Script



The screenshot shows a script editor window with three tabs: "woprnt.rptdesign", "asset_costrullup_update.rptdesign", and "asset_costrullup.rptdesign". The "Script:" dropdown is set to "open". A dropdown menu is open over the "fetch" script, showing options: "open", "describe", "fetch", "close", and "beforeOpen". The "fetch" script is selected. The script content is as follows:

```
1 open
2 describe
3 fetch
4 close
5 beforeOpen
6
7 // Add query to sqlText variable.
8 sqlText = "select workorder.wonum, workorder.taskid, workorder.istask, matusetrans.itemnum, matusetrans.description,
9 + "matusetrans.location, matusetrans.quantity, matusetrans.unitcost, matusetrans.linecost, matusetrans.issueType "
10 + "from matusetrans, workorder "
11 + "where workorder.siteid = '" + rows[0]["siteid"] + "'"
12 + "and workorder.wonum = matusetrans.refwo "
13 + "and ((workorder.parent = '" + rows[0]["wonum"].replace('/','g','') + "' and workorder.taskid is not null) "
14 + "or (workorder.wonum = '" + rows[0]["wonum"].replace('/','g','') + "' and workorder.taskid is null)) "
15 + "and workorder.siteid = matusetrans.tositeid "
16 + "order by workorder.taskid"
17 ;
18 actualMaterialsDataSet.setQuery(sqlText);
```

C6. Add “ row["issueType"] = actualMaterialsDataSet.getString("issueType");” before “return (true)” statement

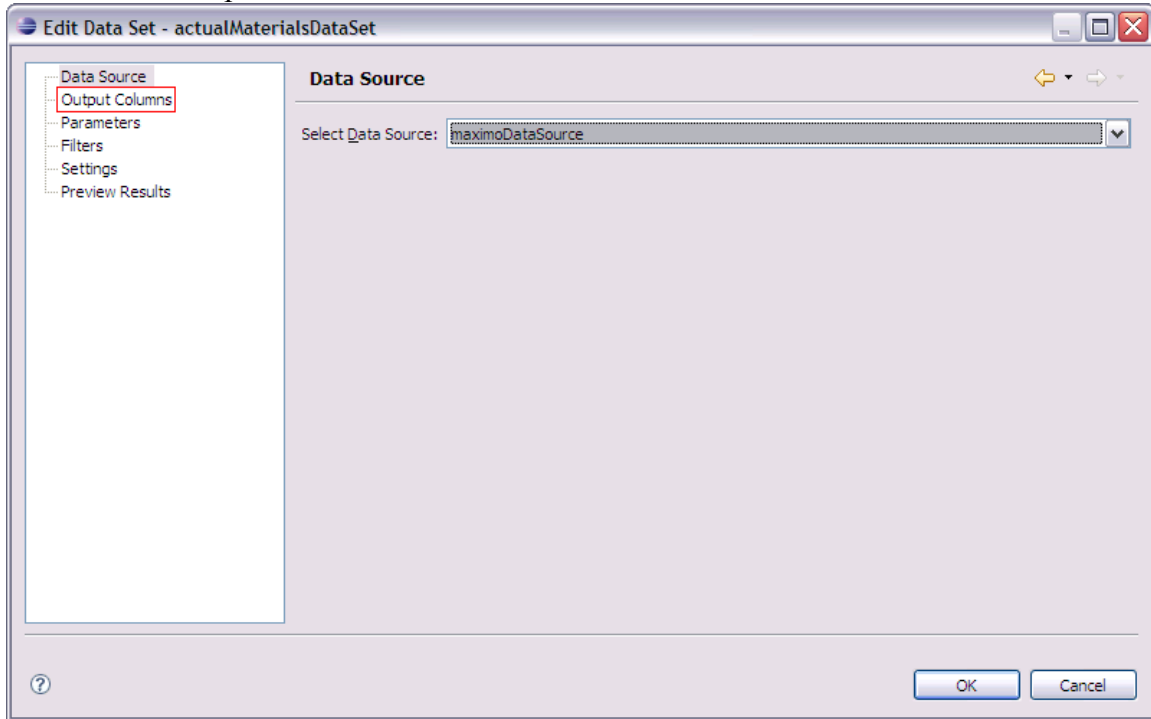


The screenshot shows the same script editor window. The "Script:" dropdown is now set to "fetch". The script content is as follows:

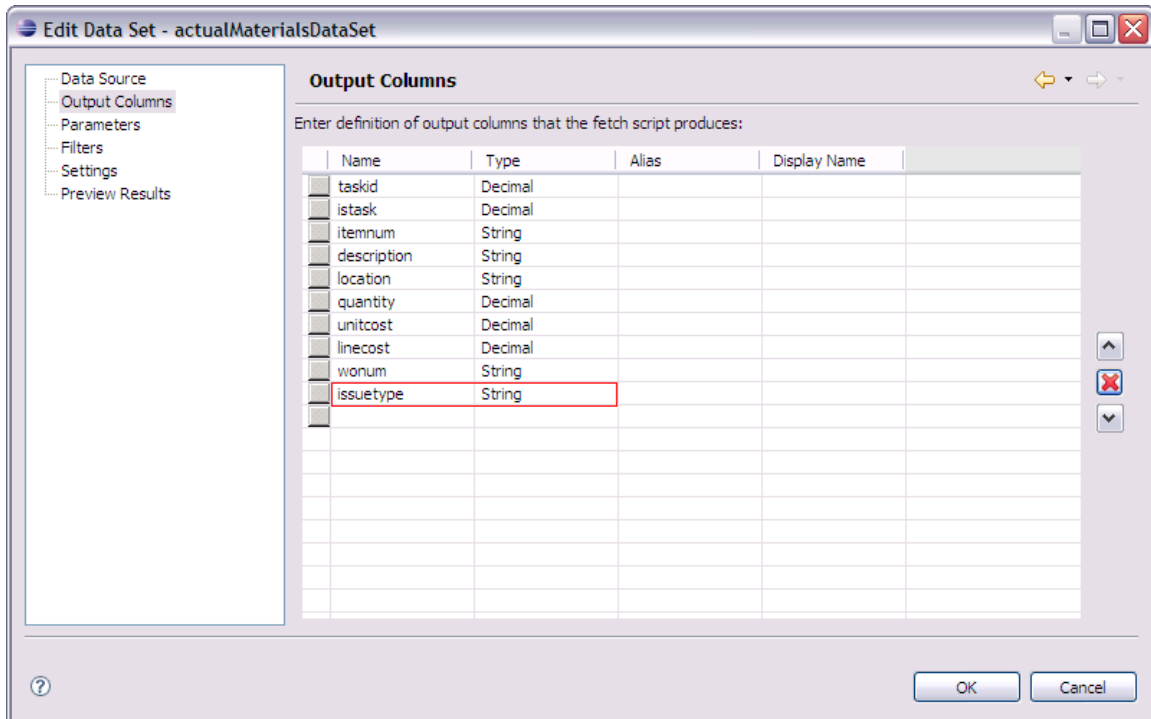
```
1 if (!actualMaterialsDataSet.fetch())
2     return (false);
3
4 // Add a line for each output column
5 // The specific get method should match the data type of the output column.
6
7 row["wonum"] = actualMaterialsDataSet.getString("wonum");
8 row["itemnum"] = actualMaterialsDataSet.getString("itemnum");
9 row["description"] = actualMaterialsDataSet.getString("description");
10 row["location"] = actualMaterialsDataSet.getString("location");
11 row["taskid"] = actualMaterialsDataSet.getDouble("taskid");
12 row["istask"] = actualMaterialsDataSet.getDouble("istask");
13 row["linecost"] = actualMaterialsDataSet.getDouble("linecost");
14 row["unitcost"] = actualMaterialsDataSet.getDouble("unitcost");
15 if (actualMaterialsDataSet.getDouble("quantity") < 0)
16     row["quantity"] = actualMaterialsDataSet.getDouble("quantity") * -1;
17 else
18     row["quantity"] = actualMaterialsDataSet.getDouble("quantity");
19 row["issueType"] = actualMaterialsDataSet.getString("issueType");
20
21 return (true);
```

C7. On Data Explorer view, double click actualMaterialsDataSet Data Set

C8. Click on Output Columns



C9. Add: Name: issuetype
Type: string
Click on OK button



C10. Click on Layout tab

```

1 if (!actualMaterialsDataSet.fetch())
2     return (false);
3
4 // Add a line for each output column
5 // The specific get method should match the data type of the output column.
6
7 row["wonum"] = actualMaterialsDataSet.getString("wonum");
8 row["itemnum"] = actualMaterialsDataSet.getString("itemnum");
9 row["description"] = actualMaterialsDataSet.getString("description");
10 row["location"] = actualMaterialsDataSet.getString("location");
11 row["taskid"] = actualMaterialsDataSet.getDouble("taskid");
12 row["istask"] = actualMaterialsDataSet.getDouble("istask");
13 row["linecost"] = actualMaterialsDataSet.getDouble("linecost");
14 row["unitcost"] = actualMaterialsDataSet.getDouble("unitcost");
15 if (actualMaterialsDataSet.getDouble("quantity") < 0)
16     row["quantity"] = actualMaterialsDataSet.getDouble("quantity") * -1;
17 else
18     row["quantity"] = actualMaterialsDataSet.getDouble("quantity");
19 row["issuetype"] = actualMaterialsDataSet.getString("issuetype");
20
21 return (true);

```

Layout Master Page Script XML Source Preview

C11. On the report layout, click on Actual Materials label field.

Task ID	Craft	Skill Level	Labor	Vendor	Contract Num	Regular	Premium Hours	Regular	Premium Rate	Line Cost
[taskid]	[craft]	[skilllevel]	[laborcode]	[vendor]	[contractnum]	[regularhrs]	[premiumpaghours]	[pagrate]	[premiumpagrate]	[linecost]
Total Actual Labor:										[data item]

Task ID	Item	Description	Storeroom	Qty	Unit Cost	Line Cost
[taskid]	[itemnum]	[description]	[location]	[quantity]	[unitcost]	[linecost]
Total Actual Materials:						[data item]

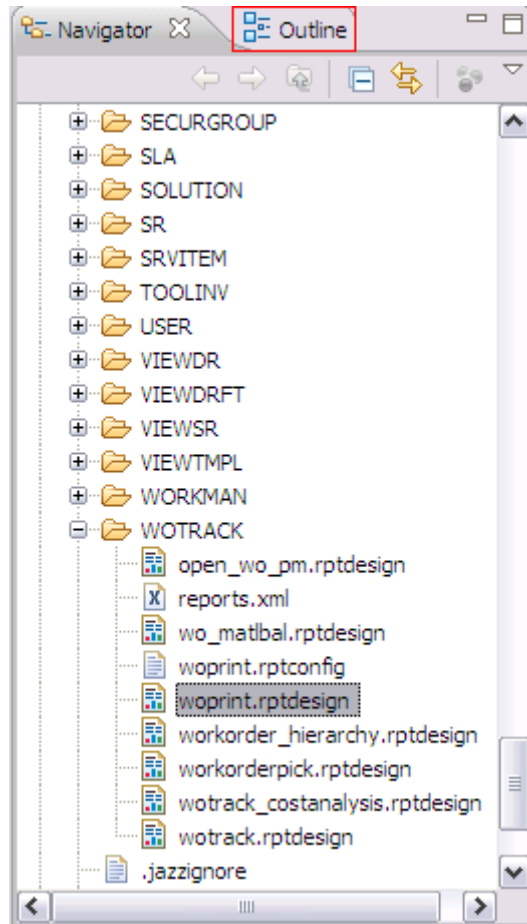
Task ID	Service Item	Description	Vendor	Qty	Unit Cost	Line Cost
[taskid]	[itemnum]	[description]	[vendor]	[quantity]	[unitcost]	[linecost]
Total Actual Services:						[data item]

Task ID	Tool	Description	Qty	Hrs	Rate	Line Cost
[taskid]	[itemnum]	[description]	[toolqty]	[toolhrs]	[toolrate]	[linecost]
Total Actual Tools:						[data item]

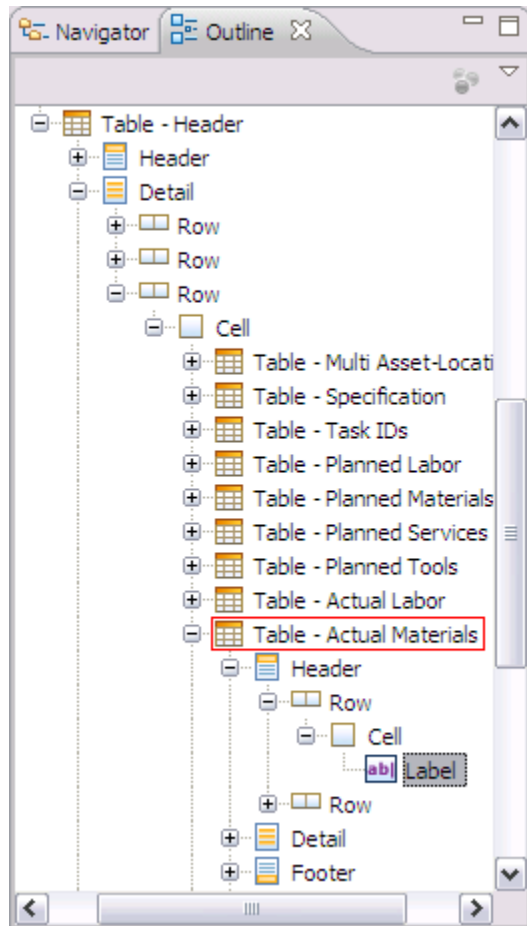
WO	Description	Class	Status	Relationship
[relatedrekey]	[description]	[relatedreclass]	[status]	[relatetype]
Ticket	Description	Class	Status	Relationship
[ticketid]	[description]	[relatedreclass]	[status]	[relatetype]

Layout Master Page Script XML Source Preview

C12. Click on Outline tab.



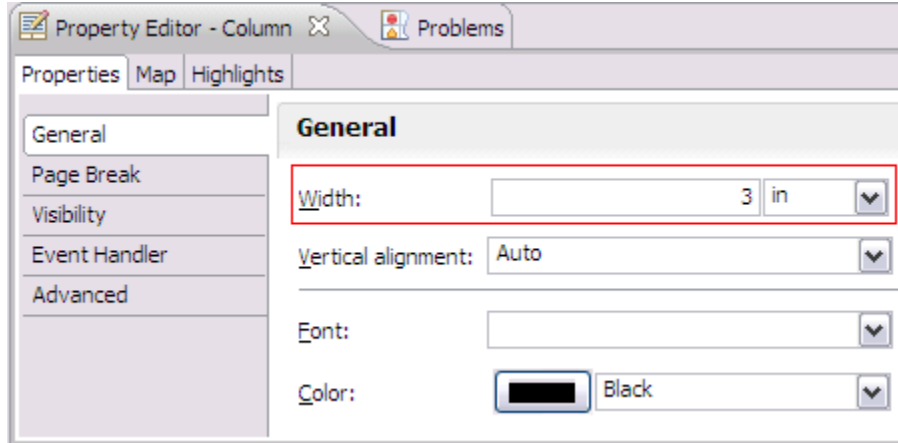
C13. Select Table – Actual Materials



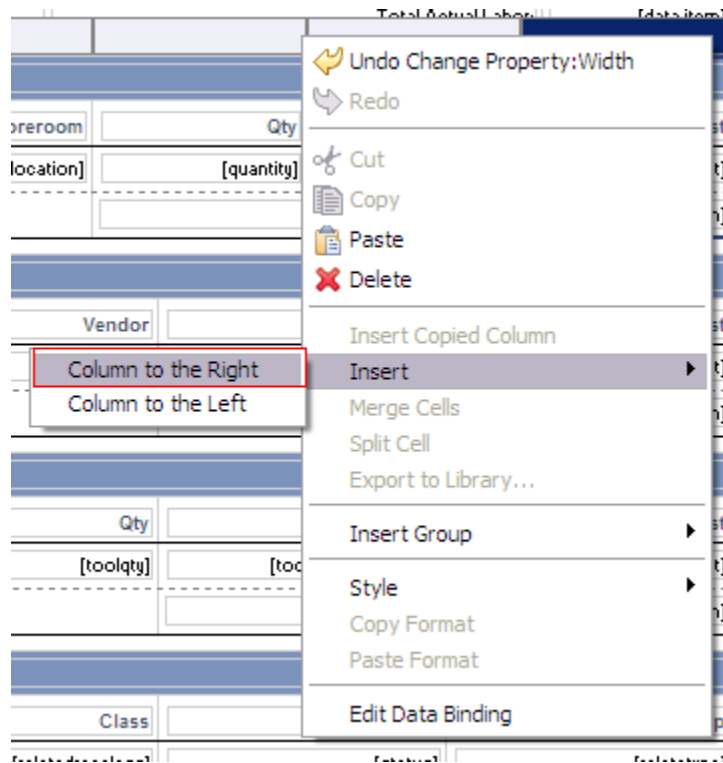
C14. Select the top of Description column

[taskid]	[ortart]	[skillevet]	[laborcode]	[vendor]	[contractnum]	[regulartms]	[premiumpagnours]	[pagrate]	[premiumpagrate]	Total Actual Labor
Actual Materials										
Task ID	Item	Description			Storeroom	Qty	Unit Cost	L		
[taskid]	[itemnum]	[description]			[location]	[quantity]	[unitcost]	[c		
Footer Row										
							Total Actual Materials:		[c	

C15. On Property Editor, change column width to 3



C16. Right click on the top of Line Cost column and click on Insert > Column to the Right



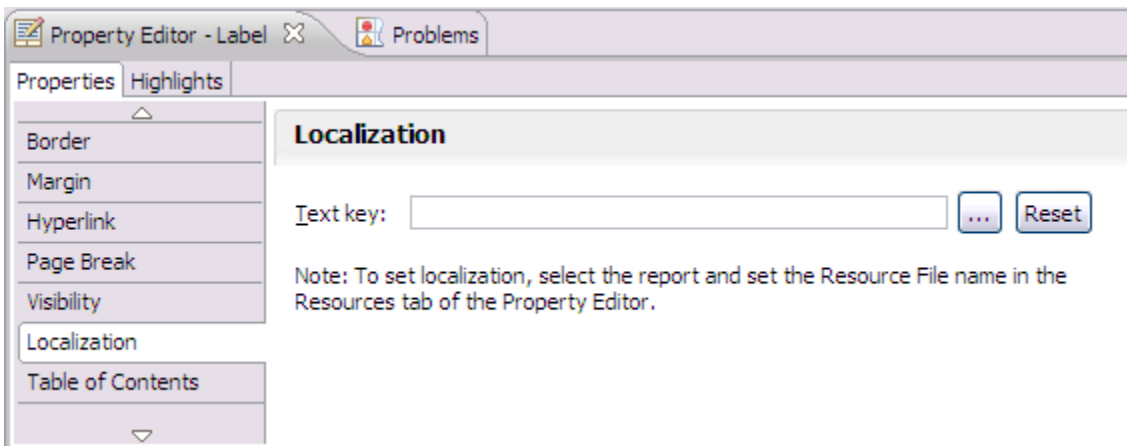
C17. Select both Actual Materials cell and the cell in blue you've just created.

Task ID	Item	Description	Storeroom	Qty	Unit Cost	Line Cost
[taskid]	[itemnum]	[description]	[location]	[quantity]	[unitcost]	[linecost]
Footer Row					Total Actual Materials:	[data item]

C18. Right click on the blue empty cell you've selected and click on Merge Cells

C19. Right click on the line next to Line Cost field. Click on Insert > Label

C20. On Property Editor, Select Localization tab and click on "..." button

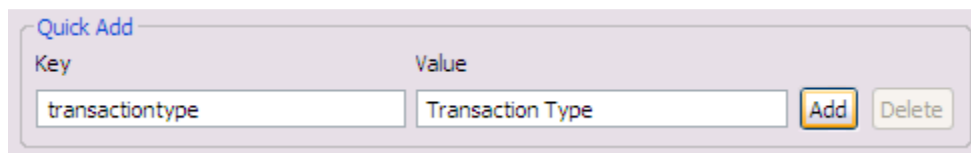


C21. Enter:

Key: transactiontype

Value: Transaction Type

Click on Add button



C22. Select transactiontype on the list and click on OK button.

C23. Expand actualMaterialDataSet from Data Explorer view.

C24. Drag and drop `issuetype` from `actualMaterialDataSet` to the report

Qty		Unit Cost	Line Cost	Transaction Type
[quantity]	[unitcost]	[linecost]	[issuetype]	
Total Actual Materials:			[data item]	

C25. The new column should look like this:

Qty	Unit Cost	Line Cost	Transaction Type
[quantity]	[unitcost]	[linecost]	[issuetype]
Total Actual Materials:		[data item]	

C26. Save the report

After your report modifications are complete, import the updated report. You can do this either via the command utility, or thru the action available in Maximo's Report Administration application.

For additional details on importing, reference the V7 Report Feature Guide available on IBM's support site.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “[Copyright and trademark information](#)” at www.ibm.com/legal/copytrade.shtml.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both