lucity™

TRAINING GUIDE

Crystal Reports

Advanced 2

Using Crystal Reports with Lucity

Advanced – 2

The sixth of a seven-part series, this workbook is designed for Crystal Reports® users with some experience. Here, you'll learn to set up Hide (Drill-Down), Variables, Percentages and Commands. Issues with Web and Dashboard reports will be addressed.

Table of Contents

Hide (Drill-Down)

Sometimes, it's useful to view a condensed version of a report, where the details are accessible only when needed. This is often helpful when trouble shooting bad data. This option is available when grouping data. The following example will show you how to display the number of Work Orders for each category with the option to view the individual Work Orders within the categories.

The main report looks like this:

Work Orders by Category Report *8/27/2012*
8:11 AM

Double click the category code to view the individual work orders.

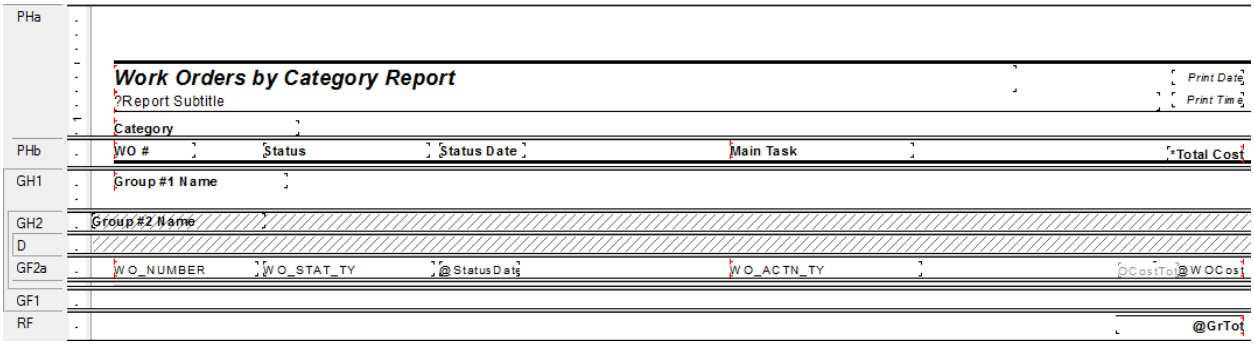
Category	Count
13310 Auxiliary Equipment	1
41310 Backflow Preventors	1
12100 Bridge Maintenance	1
12510 Bus Stops	1
13320 Cabinet	1
02000 Call Center	3
92100 Commercial Collecti	2
13341 Conductor Cabling	1
13342 Conduit Cabling	1
12520 Curb	1
11300 Detention Basins	1
60000 Equipment	99

The hidden section of the Equipment category looks like this:

Equipment

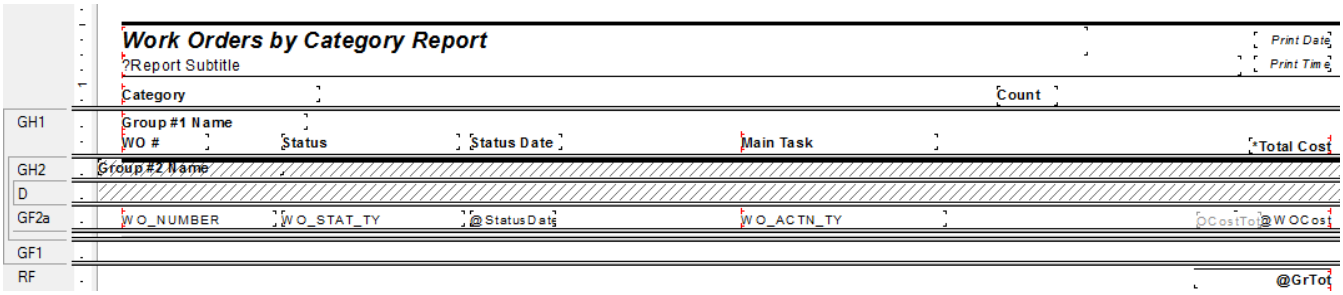
WO #	Status	Status Date	Category	Main Task	*Total Cost
2006-01662	Complete	7/25/2006	Equipment	Miscellaneous	$156.59
2006-01682	Complete	8/7/2006	Equipment	Dist Maintenance/Equipment	$155.47
2006-01684	Complete	8/7/2006	Equipment	Miscellaneous	$208.35
2006-01685	Complete	8/7/2006	Equipment	Miscellaneous	$199.74

1. We will modify the **Work Order Summary Report** (**WOSum.rpt**) from the Work Order module. From the Work report folder open **LC_WOHide.rpt**.
2. Change the title to **Work Orders by Category Report**.
3. Delete the **Address** column.
4. Add the **Category** grouping and move the Category grouping to be the first grouping.
5. Add a second *Page Header* section above the Column titles by placing the cursor on the left ruler section, click and wiggle to create the break.
6. Move the **Category** title above **WO #** and remove the Category field.
7. Increase the *Group Header 1* section size.

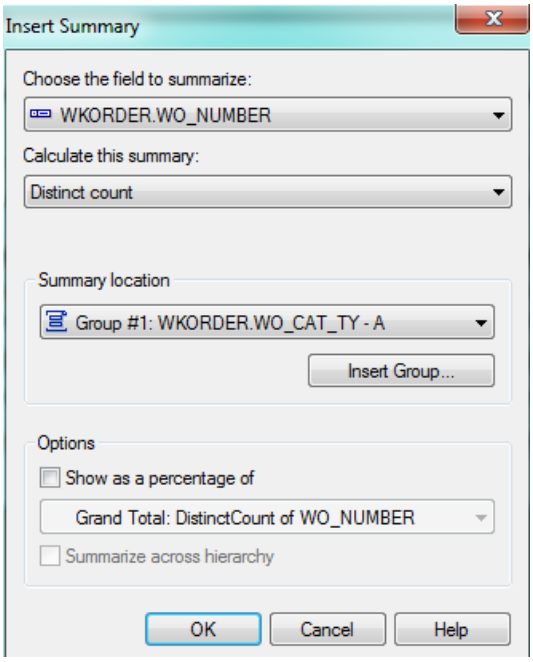


8. Right click in the *Page Header b* section, click *Select All Section Objects*.

9. Drag the column titles into the *Group Header 1* section, beneath **Group #1 Name**.
10. Right click in the *Page Header a* section and select *Merge Section Below*.
11. Add a column title called **Count** in the *Page Header* section.



12. Create a count formula using the *Summary* button Σ to count the number of Work Orders (*Distinct Count* to deal with any filtering issues).
    a. Have the *Count* formula placed in the ***Group Footer 1*** section.
    b. Move the formula to the **Count** column.

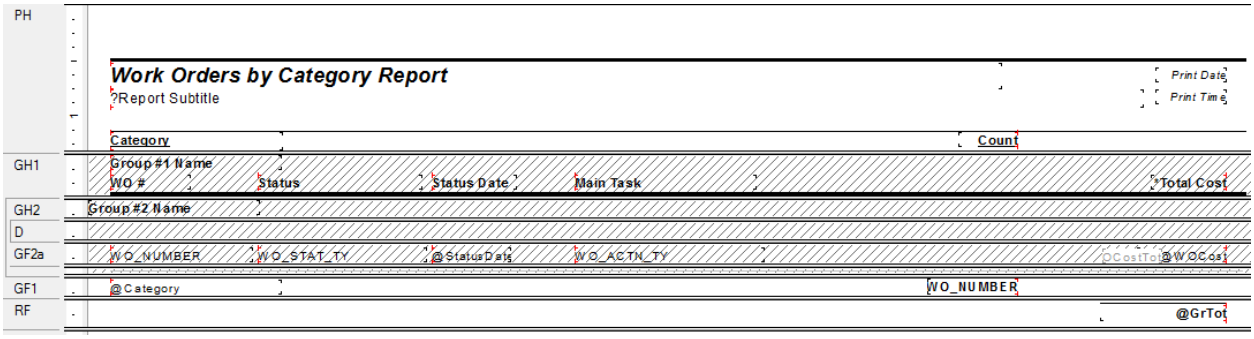


13. Next, we will create a concatenated formula where fields can be "tacked together" to eliminate awkward spacing. Create a new formula called **Category**.

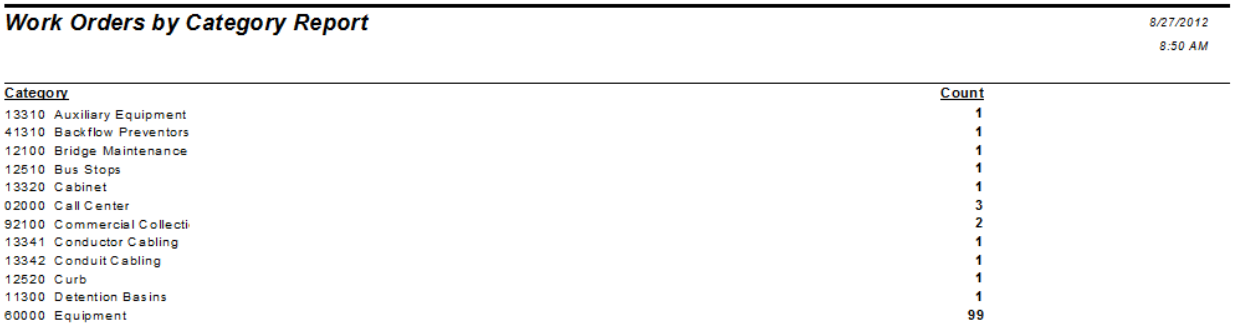
```
if isnull({WKORDER.WO_CAT_CD}) then "" else {WKORDER.WO_CAT_CD}&"  "&
if isnull({WKORDER.WO_CAT_TY}) then "" else {WKORDER.WO_CAT_TY}
```

14. Drag this formula into the *Group Footer 1* section under the **Category** column title.

15. To the left of the report, right click in the *Group Header 1* section and select *Hide(Drill-Down)*.
16. Repeat step 14 for the *Group Footer 2a* section.



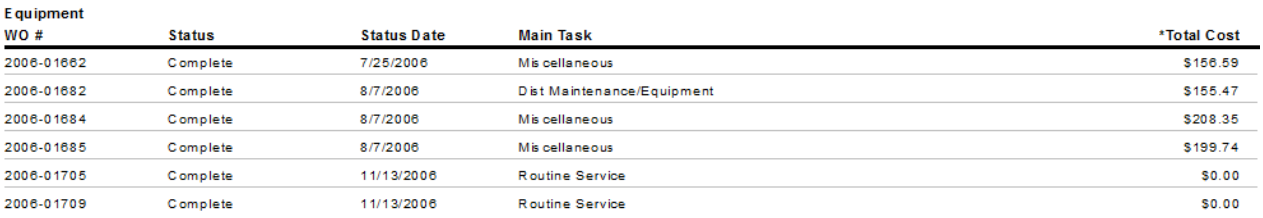
17. Preview the report to see how it looks.



Work Orders by Category Report 8/27/2012 8:50 AM

Category	Count
13310 Auxiliary Equipment	1
41310 Backflow Preventors	1
12100 Bridge Maintenance	1
12510 Bus Stops	1
13320 Cabinet	1
02000 Call Center	3
92100 Commercial Collecti	2
13341 Conductor Cabling	1
13342 Conduit Cabling	1
12520 Curb	1
11300 Detention Basins	1
60000 Equipment	99

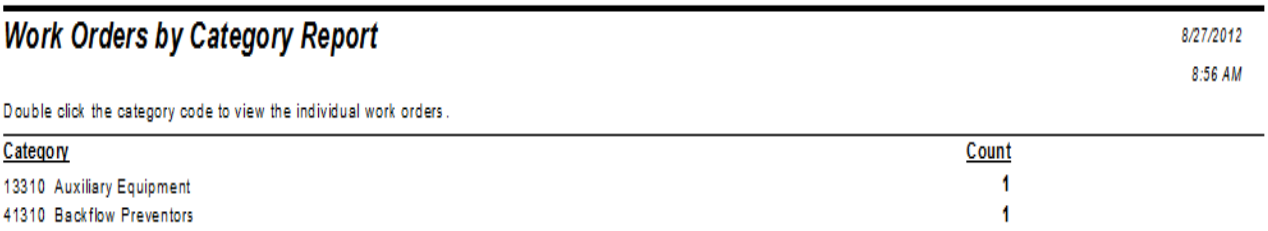
- To view the Work Orders within each category, double click the category code.



Equipment

WO #	Status	Status Date	Main Task	*Total Cost
2006-01662	Complete	7/25/2006	Miscellaneous	$156.59
2006-01682	Complete	8/7/2006	Dist Maintenance/Equipment	$155.47
2006-01684	Complete	8/7/2006	Miscellaneous	$208.35
2006-01685	Complete	8/7/2006	Miscellaneous	$199.74
2006-01705	Complete	11/13/2006	Routine Service	$0.00
2006-01709	Complete	11/13/2006	Routine Service	$0.00

18. Next, you should add a text line in the *Page Header* that instructs the report viewer to double click on the category code to view the individual work orders.



Work Orders by Category Report 8/27/2012 8:56 AM

Double click the category code to view the individual work orders.

Category	Count
13310 Auxiliary Equipment	1
41310 Backflow Preventors	1

Example 27A

Variables

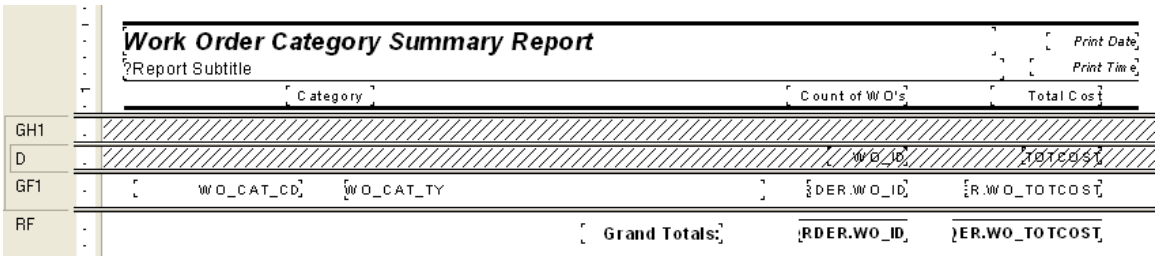
Why Use Variables?

One of the more important uses of variables is in reports that may be run with a filter of a "child" or "grid" field that could have multiple values. Other uses are in running subtraction calculations in progressing records or passing information from a subreport to the main report. There are many uses that only become apparent when the use of the standard formulas doesn't work.

A variable is a type of component that may be used in a formula. A variable represents a specific item of data or a value. It then acts as a placeholder for that value. When a formula encounters a variable, it searches for the variable's value and uses it in the formula.

Unlike a constant value, which is fixed and unchanging, a variable can repeatedly be assigned different values. When you assign a value to a variable, it maintains that value until you later assign it a new value. Because of this flexibility, it is necessary for you to declare variables before you use them. It is also frequently necessary to reset the variable.

The original **Work Order Category Summary Report** (**WOCatSum.rpt**) looked like this:



It was a very simple report which grouped on Category and used the Crystal Summary tool to calculate the Group Total and Grand Total. This worked when the report was run without filters.

3010	Tree	2	$714.51
40000	Water Department	26	$464.08
41000	Potable	44	$2,596.80
41110	Potable Mains	124	$20,068.40

There were certain filters that caused duplicate Work Order costs. The problem filters are the fields that come from grid data. In the Work Order module these would include Location, Assets, Tasks and Resources. This occurs because of the filter statement being passed from Lucity to Crystal. If there are two Tasks on a Work Order that are true for a Task filter being run, then the report will run the record twice.

When the report was run with a resource filter on Work Orders that have employees, these same categories looked like this:

3010	Tree	2	$2,143.54
40000	Water Department	25	$464.08
41000	Potable	40	$10,403.04
41110	Potable Mains	122	$59,106.27

There is an obvious difference in the cost fields. Instead of possibly going down due to records being filtered out that had no employees, the cost actually went up. Some work orders had multiple employees so the cost was summed each time the filter was true. The count was calculated using the Distinct Count option as opposed to the Count option for calculating so there was no duplication of records.

To correct this issue we used grouping and variables.

3010	Tree	2	$714.51
40000	Water Department	25	$464.08
41000	Potable	40	$2,596.80
41110	Potable Mains	122	$20,068.40

Declaring a Variable

Each variable must be assigned a **data type** (string, number, currency, time, or date) and a **name** (your choice). You also need to know the **scope** or degree to which the variable will be used:

- **Local** - Variable will be used in a single formula.
- **Global** - Variable will be used throughout the main report. This is Default if the scope isn't stated.
- **Shared** - Variable will be used throughout the main report and any subreports.

1. To declare a variable, create a formula using the *Formula Workshop* or type it in.
   - Within the *Formula Workshop,* there are two options under ***Operators*** called ***Scope*** and ***Variable Declarations***.
   - Open *Scope* and double click the option that you want (typically – ***Shared***).
   - Open *Variable Declarations* which lists the correct format used in stating the *Data Type*. Double click the correct *Data Type*.
   - The correct format requires that the scope be placed before the Data Type.

     Shared numberVar := ;

     - For just declaring a variable, you should remove the colon and equal sign ":=".

   - The name for the variable goes after the Data Type.

     <Scope> <Data Type><VariableName>;

     Example: **Shared NumberVar Security;**

     In our example, the Scope is **Shared** and the Data Type is **NumberVar**. The Variable name is **Security**.
2. After the formula is created it needs to be placed in the report.

Assigning a Value to a Variable

After a variable has been declared, a value can be assigned to it. The assigned value can be a direct value, formula, parameter, or the value of a field.

<VariableName> := <Value>

Example: **Security := {?ViewSecuredFields};**

Declare and Assign

You can also declare a variable and assign a value to it in one step. In this case you would not have removed the ":=".

Example: **Shared NumberVar Security := {?ViewSecuredFields};**

Evaluation Time

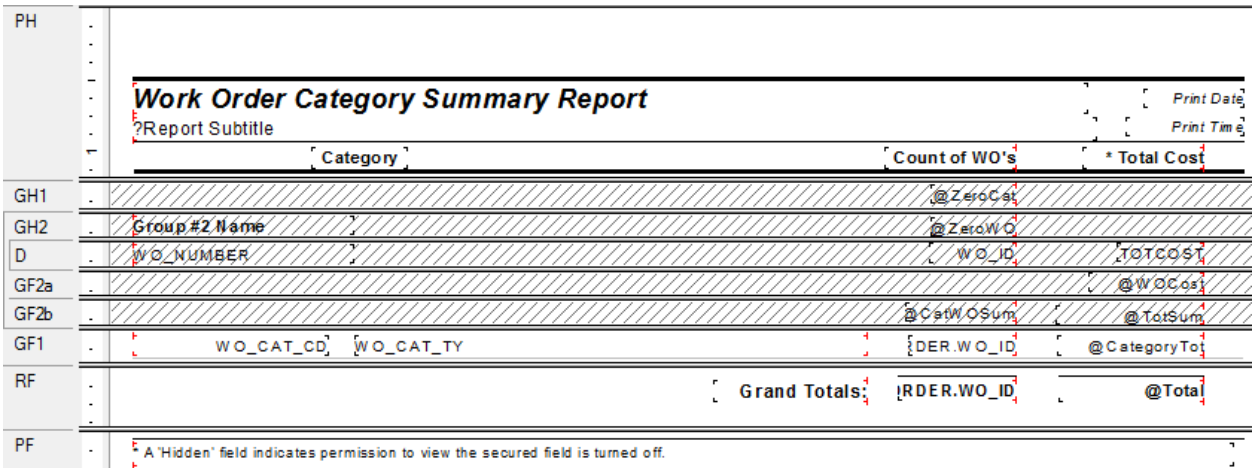
Evaluation Times are statements that tell when the formula is to be evaluated. If nothing is stated, Crystal Reports guesses what is appropriate for the data being used in the formula. Within the *Formula Workshop* these options can be found under ***Functions > Evaluation Time***.

- **BeforeReadingRecords**: formulas are evaluated before the database records are read.
- **WhileReadingRecords**: formulas are evaluated while it is reading the database records.
- **WhilePrintingRecords**: formulas are evaluated while it is printing the database record data.
- **Evaluate After (x)**: formula forces this formula to calculate after the "x" formula.

Typically **WhilePrintingRecords** is used.

Using Variables in a Report

One way to correct the **WOCatSum.rpt** report would be to use variables.



1. Add a second grouping on the Work Order Number, **WO_NUMBER**.

2. Create a formula for the Work Order Cost (**WOCost**) and place it in the WO Number *Group Footer 2*.

```
WhilePrintingRecords;
Shared numberVar WOCost ;
WOCost:={WKORDER.WO_TOTCOST}
```

3. Create a formula to summarize the cost for the Category (**CatWOSum**). Add a new section below the WO Number *Group Footer* 2. This is a second *Group Footer* (*GF2b*) for this section. Place this new formula in this section.

```
WhilePrintingRecords;
Shared numberVar CatTot ;
Shared numberVar WOCost ;
CatTot:= CatTot + WOCost
```

4. Create a formula to summarize the total cost for all Work Orders (**TotSum**). Place this in *GF2b*.

```
WhilePrintingRecords;
Shared numberVar TotWOCost ;
Shared numberVar WOCost ;
TotWOCost:= TotWOCost + WOCost
```

5. Create a formula to reset or zero the Work Order cost variable (**ZeroWO**). Place this in the WO Number *Group Header #2* section.

```
WhilePrintingRecords;
Shared numberVar WOCost:=0;
```

6. Create a formula to reset the Category Cost variables (**ZeroCat**). Place this in the Category *Group Header #1*.

```
WhilePrintingRecords;
Shared numberVar CatTot:=0;
```

7. Create a formula to show the Category total cost (**CategoryTot**). Place this in the Category *Group Footer #1*.

```
WhilePrintingRecords;
Shared numberVar CatTot ;
CatTot
```

8. Create a formula to show the Grand Total Cost (**Total**). Place this in the *Report Footer*.

```
WhilePrintingRecords;
Shared numberVar TotWOCost ;
TotWOCost
```

9. Suppress the new *Group Header* and *Footer* sections.

Example 28A

It depends on the report as to whether you use variables or running totals to solve filter issues. At the time the report was revised the use of variables was the choice method to correct the issue.

Variables are also very useful to pass information from a subreport to the main report. This was shown in **Unlinked Subreports** mentioned in **Intermediate 2**. This is shown again here now that variables have been discussed.

Unlinked Subreports

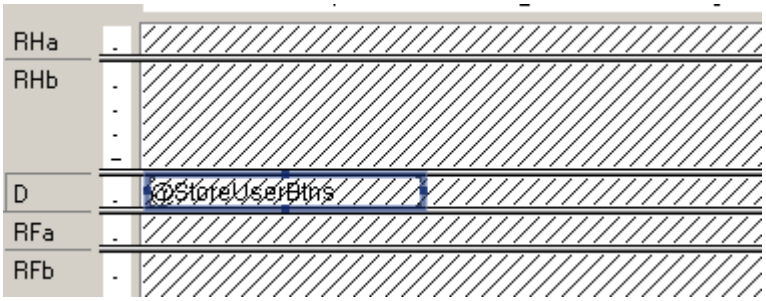
There are times when the subreports do not need to be linked. For example, you won't need to link subreports if there are no connections to the main report. In this situation, the subreport can share data using variables but does not necessarily share a database.

Stand-alone reports can be brought in as subreports so that one report can bring in many reports or just the final values of these reports as a summary.

In *Lucity* software the button captions may be altered to suit individual needs. These field captions may be reflected in the report by one of the following methods:

- Edit the appropriate text object in the report as discussed in the related Beginning Crystal workbooks.
- Pull the value that is stored in the database and display it on the report using an unlinked subreport. This is the setup for the User fields in the Custom tab.

Lucity uses this type of unlinked subreport in many of the Detail reports. The subreport is hidden in a Report Header. It contains a formula declaring variables that are associated with the User button captions in the Custom tab.



The **@StoreUserBtns** formula appears on the following page. Note that this is only part of the formula. We've included this to make you aware of what happens behind the scenes.

- Any line starting with // is just a comment; it is not part of the formula.

The formula for each User button caption looks something like this:

```
WhilePrintingRecords;
Shared StringVar strUser1;
strUser1
```

```
//The subreport that this formula belongs to must contain the following
//Select Expert statement:
//{WTFIELDS.FieldName} like ["BD_USE*"]
//The Fields table reference should match your database.  The two character
//data field prefix and suffix must be changed for each report.
//You must also enter your specific Fields Table name in the "IF" statements
//found below (e.g. - WTFIELDS):
WhilePrintingRecords;
//Enter two-character field prefix here (e.g. - 'BD'):
StringVar ModPrefix:='BD';
StringVar strUser1Field:=ModPrefix+'_USER1CD';
StringVar strUser2Field:=ModPrefix+'_USER2CD';
StringVar strUser3Field:=ModPrefix+'_USER3CD';
StringVar strUser4Field:=ModPrefix+'_USER4';
StringVar strUser5Field:=ModPrefix+'_USER5';
StringVar strUser6Field:=ModPrefix+'_USER6';
StringVar strUser7Field:=ModPrefix+'_USER7';
StringVar strUser8Field:=ModPrefix+'_USER8';
StringVar strUser9Field:=ModPrefix+'_USER9';
StringVar strUser10Field:=ModPrefix+'_USER10';
StringVar strUser11Field:=ModPrefix+'_USER11';
StringVar strUser12Field:=ModPrefix+'_USER12';
StringVar strUser13Field:=ModPrefix+'_USER13';
StringVar strUser14Field:=ModPrefix+'_USER14';
StringVar strUser15Field:=ModPrefix+'_USER15';
Shared StringVar strUser1;
Shared StringVar strUser2;
Shared StringVar strUser3;
Shared StringVar strUser4;
Shared StringVar strUser5;
Shared StringVar strUser6;
Shared StringVar strUser7;
Shared StringVar strUser8;
Shared StringVar strUser9;
Shared StringVar strUser10;
Shared StringVar strUser11;
Shared StringVar strUser12;
Shared StringVar strUser13;
Shared StringVar strUser14;
Shared StringVar strUser15;
If {WTFIELDS.FieldName} = strUser1Field then
    If not IsNull ({WTFIELDS.UserName}) and {WTFIELDS.UserName} <> '' then
    strUser1:={WTFIELDS.UserName}+':'
    Else strUser1:={WTFIELDS.DefaultName}+':';

If {WTFIELDS.FieldName} = strUser2Field then
    If not IsNull ({WTFIELDS.UserName}) and {WTFIELDS.UserName} <> '' then
    strUser2:= {WTFIELDS.UserName}+':'
    Else strUser2:= {WTFIELDS.DefaultName}+':';

If {WTFIELDS.FieldName} = strUser3Field then
    If not IsNull ({WTFIELDS.UserName}) and {WTFIELDS.UserName} <> '' then
    strUser3:= {WTFIELDS.UserName}+':'
    Else strUser3:= {WTFIELDS.DefaultName}+':';
```

Date Variables

Date fields can be converted to a string type variable that can be easier to reset.

WhilePrintingRecords;

shared stringvar FirstNoticeDue;

if {WTBFNOTICE.BN_NOT_TY}="First Notification Sent" then

FirstNoticeDue:=totext({@NoticeDt}+90,"MM/dd/yyyy")

The reset for this variable (**FirstNoticeDue**) is:

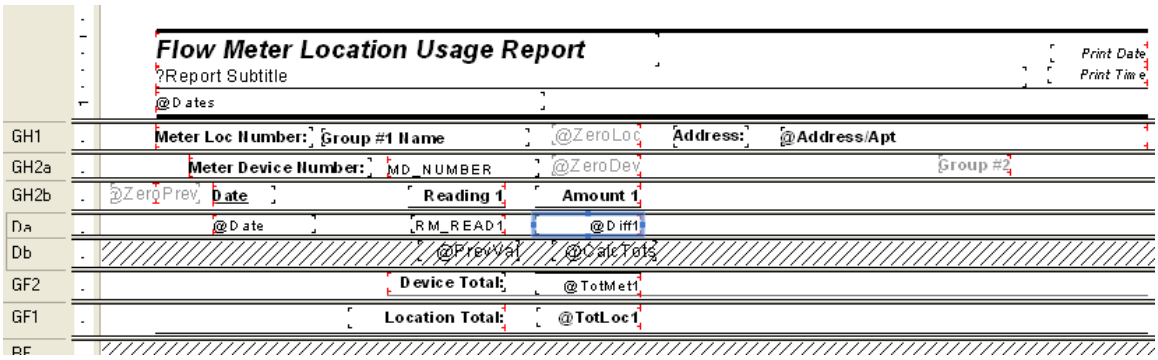
WhilePrintingRecords;

shared stringvar FirstNoticeDue:="";

Using Variables for Subtraction

Crystal Reports is quite useful when adding numbers together but it gets pretty tricky when you want to find the difference in running numbers.

In Water, the Meter Set Location module has a **Flow Meter Location Usage Report** (**WTMtUsage.rpt**) that is a good example of this. It is modified here to a single Reading to clarify the subtraction process.



The report is grouped on the **Meter Location Number (Group 1)** and then the **Device ID (Group 2)**.

- In *GH1* there is a formula to reset the Location Usage (**ZeroLoc**):

  shared Numbervar LocTot1:=0;

- In *GH2a* there is a formula to reset the Device Usage (**ZeroDev**):

  shared Numbervar MetTot1:=0;

- In *GH2b* there is a formula to reset the Device Reading 1 value (**ZeroPrev**):

  Shared Numbervar prevval1:=0;

- In *Da* there is a formula for the amount used (**Diff1**):

```
Shared Numbervar prevval1;
if PreviousIsNull ({WTMTMD.RM_READ1}) then 0
else if Previous ({WTMTMD.RM_READ1}) = 0 then 0
else if (previous ({WTMETDEV.MD_ID}) <> {WTMETDEV.MD_ID})then 0
else {WTMTMD.RM_READ1}-prevval1;
```

- In *Db* there are two formulas:

(**PrevVa**l)

```
Shared Numbervar prevval1;
prevval1:={WTMTMD.RM_READ1};
```

(**CalcTots**)

```
shared Numbervar MetTot1;
shared Numbervar LocTot1;
MetTot1:=MetTot1 + {@Diff1};
LocTot1:=LocTot1 + {@Diff1};
```

- In *GF2* there is a formula for Device total usage (**TotMet1**):

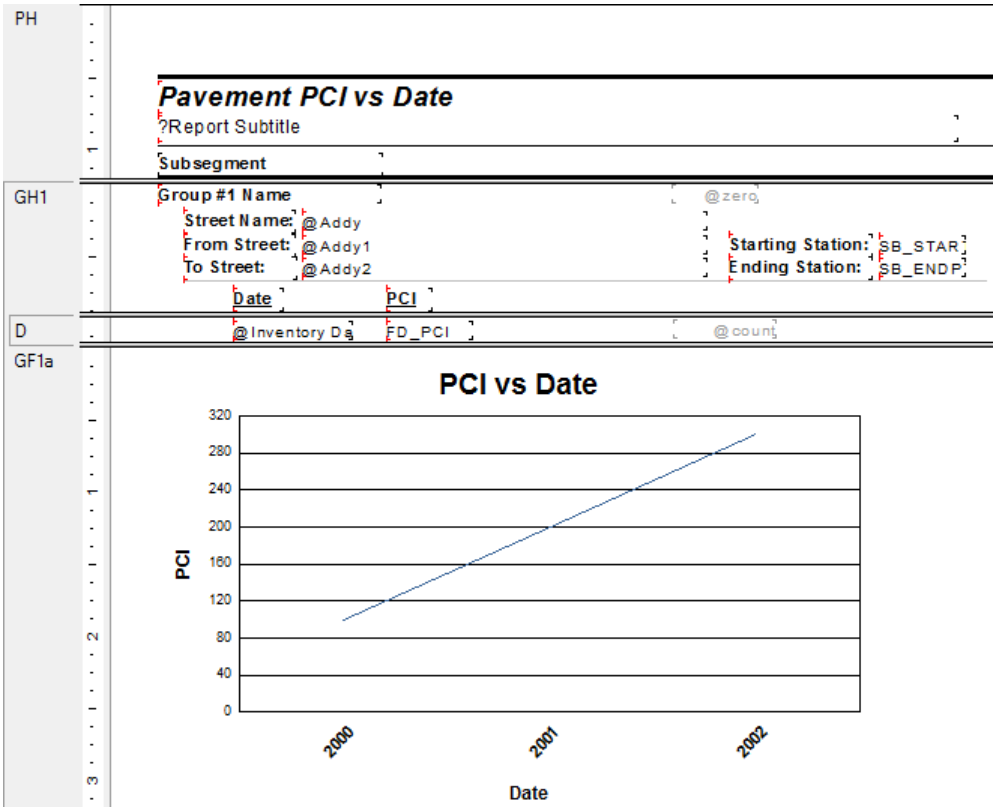
```
shared Numbervar MetTot1;
MetTot1
```

- In *GF1* there is a formula for the Location total usage (**TotLoc1**):

```
shared Numbervar LocTot1;
LocTot1
```

Using Variables to Count for Section Suppression

A report was created to view Street Subsegment information including its PCI on a given date. A graph was created for each subsegment that had more than one date. The graph needed to suppress when there were no or one date/PCI values.



The (**count**) formula was created:

WhilePrintingRecords;

Shared numberVar cnt ;

cnt:=cnt +1

Then within the *Section Expert*, *GFa1* Suppress formula was added:

WhilePrintingRecords;

Shared numberVar cnt ;

cnt in[0,1]

The section suppresses when **cnt** is equal to 0 or 1.

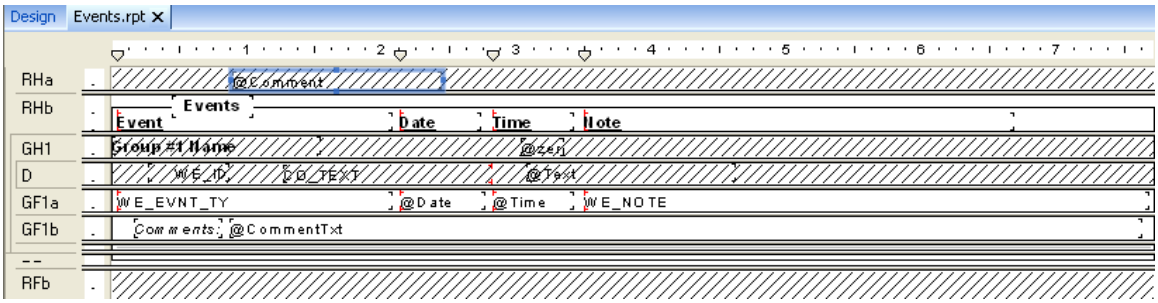
Comments to Non Comment Subreports

Subreports may not be put in subreports. Adding a Comment section that is in a separate **XXMEMO** table that requires certain criteria to be true to pull the correct data seems impossible to add to a section in a report that is already a subreport.

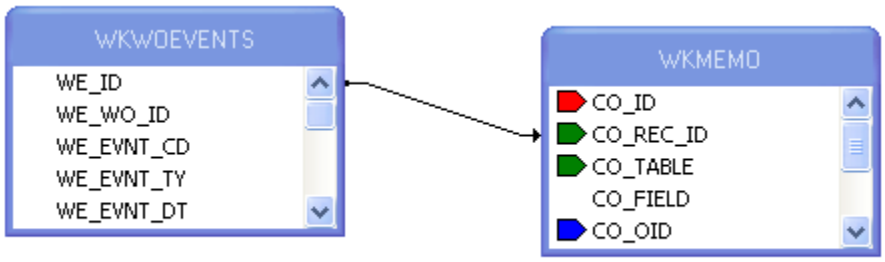
There are a variety of ways to accomplish this task. Not all work for the various scenarios.

Option 1 – Variables

A client wanted the Comment for Events added to the **Work Order Detail Report** (**WODetail.rpt**). *Events* is a subreport of this Report. It was done as follows:



1. Bring in the **WKMEMO** table and link left outer join.



2. Group on the ID of the subreport table (**WKWOEVENTS**).
3. Create a variable formula (**Text**) to grab the text of the correct Memo field.

```
WhilePrintingRecords;

Shared stringVar Text ;


If {WKMEMO.CO_FIELD}="WE_MEMO1" then

Text:={WKMEMO.CO_TEXT}
```

4. Create a reset formula (**Zero**) for this variable and place it in the *Group Header* section. The zero in this case is a space " " instead of a zero (0) because it is a string variable.

WhilePrintingRecords;

Shared stringVar Text:=" ";

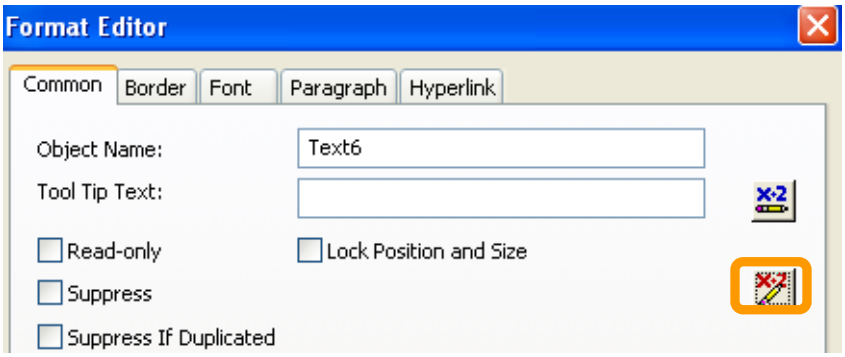
5. Place the report information in the *Group Footer* section.
6. Create another *Group Footer* section for the Comment.
7. Place the Comment text formula (**CommentTxt**) in the second footer section. Make sure this field is allowed to grow.

WhilePrintingRecords;

Shared stringVar Text ;

Text

8. For the "Comments" text field use a suppress formula when the text variable is blank.



WhilePrintingRecords;

shared stringVar Text ;

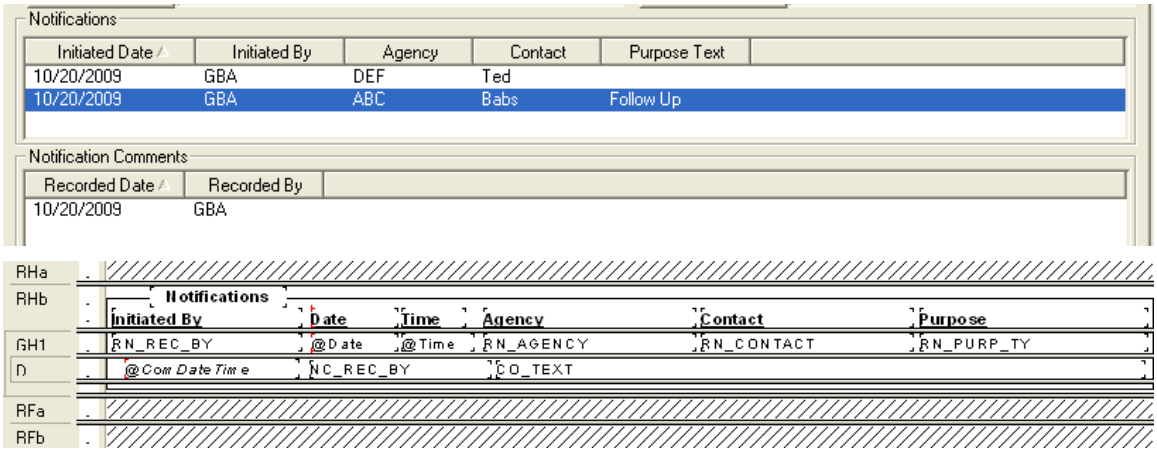
Text=" "

9. In *Section Expert* select *Suppress Blank Section* for the second *Group Footer (GF1b)*.

Option 2 – Grouping & Conditional Suppression

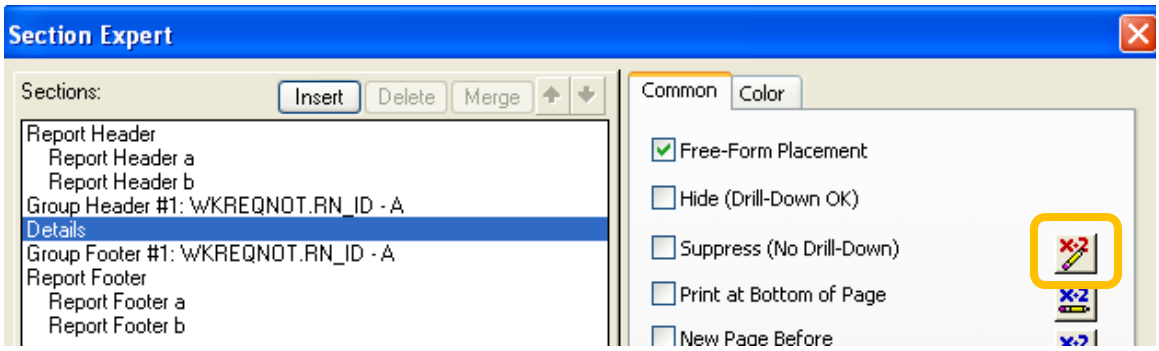
Here is another way in which a Comment was brought into a subreport.

The comment was added to the *Notifications* subreport within the **Request Detail Report** (**ReqDetail.rpt**).



1. Bring in the **WKMEMO** table and link with left outer join.

2. Group on the **WKREQNOT.RN_ID** field and place the information from this table in the *Group Header*.
3. In the *Detail* section place the information from the **WKNOTCOM** table and the **Text** field from the **WTMEMO** table.
4. In the *Section Expert*, create a conditional suppression formula for the *Details* section.



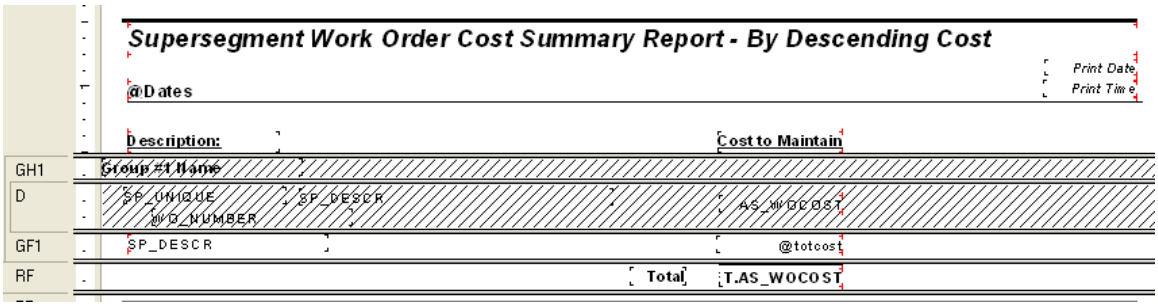
{WKMEMO.CO_FIELD} <> "NC_MEMO1"

Grouping – More Advanced

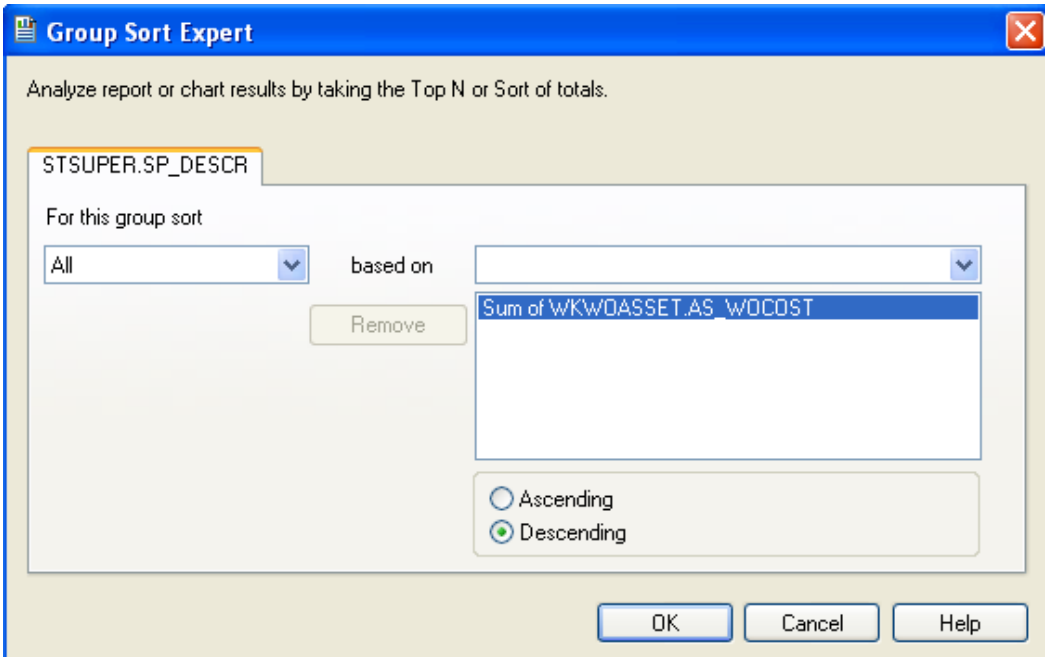
Group Sort Expert

When a report uses grouping, the report automatically sorts on the first group and then within the group any other groups. Each group can be sorted in ascending, descending or a specific order. The *Group Sort Expert*  can be used to change the field the report will sort on.

An example of this is the **Supersegment Work Order Cost Summary Report - By Descending Cost** (**SupersegDWOCost.rpt**).

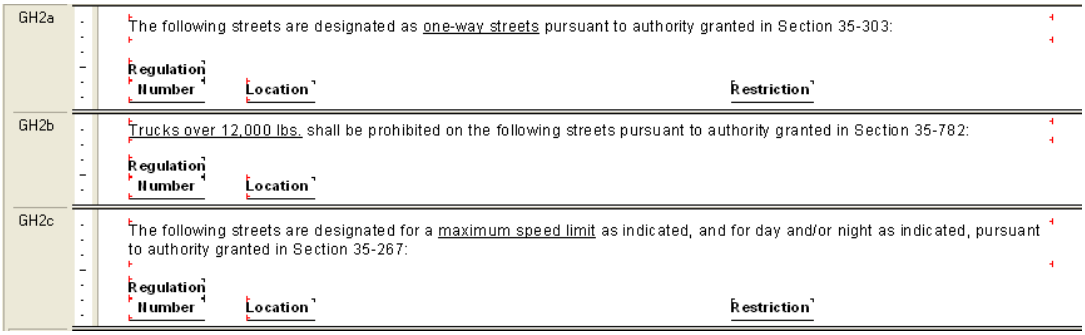


- This report was grouped on the Supersegment description field.
- The asset work order cost (**AS_WOCOST**) was being summed in the group footer.
- Within the *Group Sort Expert* the sort was changed to the cost field.



Section Expert and Grouping

If different Heading information is needed for data in the same group, you can use the *Section Expert* to create this affect.



- The *Select Expert* of the report is selecting only records where the following is true.

  {STSIGNREG.SR_REGS_CD} in [1, 2, 3 TO 12]

- Each *Group Header 2* section is grouping on **REGS_CD** in the Traffic Regulations table (**STSIGNREG**).

- Within the *Section Expert*  each *Group Header* section has a suppress formula so it will only show the records that coincide with that group heading.

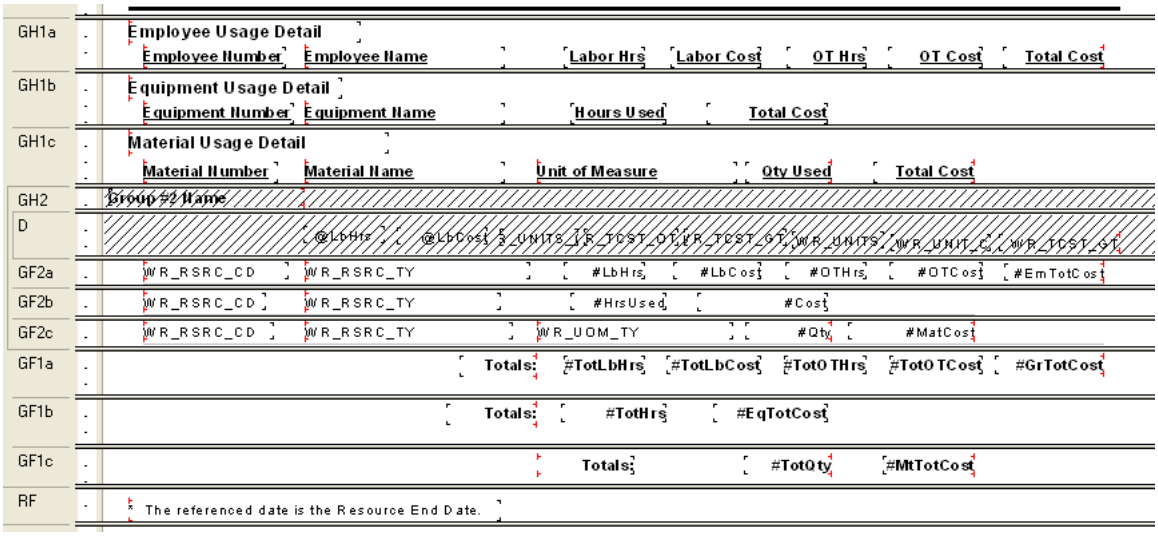
  - *GH2a* suppresses when the following is true:

    {STSIGNREG.SR_REGS_CD} in [1,3 TO 12]

    This means that *GH2a* will show up when the **REGS_CD** is equal to 2.

- The other two *GH2* sections have different suppression formulas.

Not only can you have different information in the *Group Header* sections, you can do this in the *Group Footer* as well.

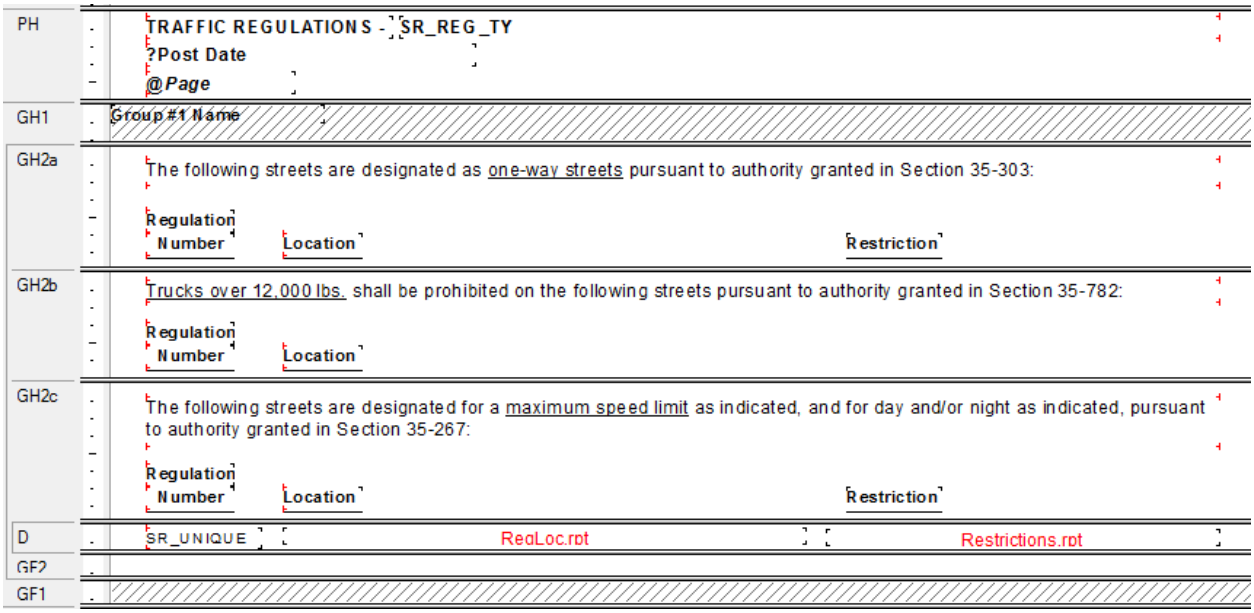


You can use the *Section Expert* to suppress the different *Header* and *Footer* sections. For example the *GH1a*, *GF2a* and *GF1a* all use the following suppression formula:

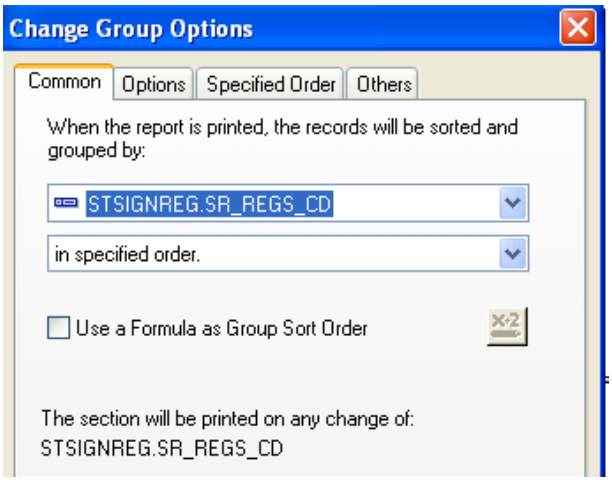
{WKRESRCE.WR_RTYP_CD} in [2,3]

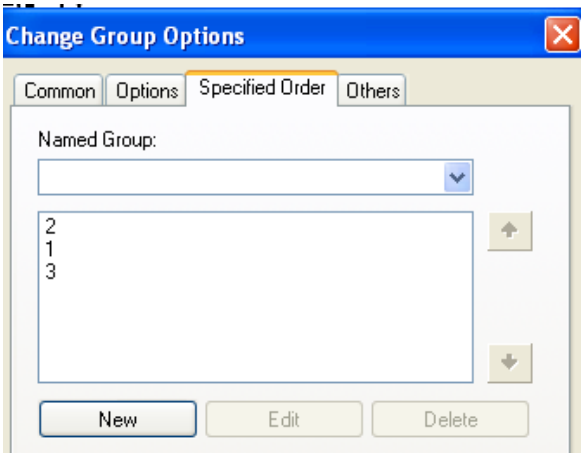
Specific Order of Grouping

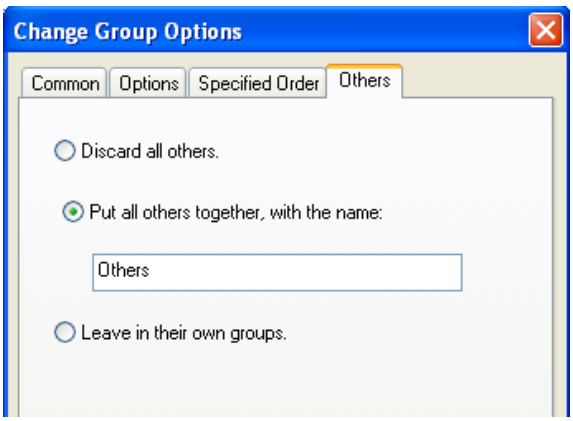
Another report using conditional suppression formulas to show different headers also needed the groups to show in a specific order.



This was done using the *Change Group* option. For *Group 2* the following option was used.







Percentages

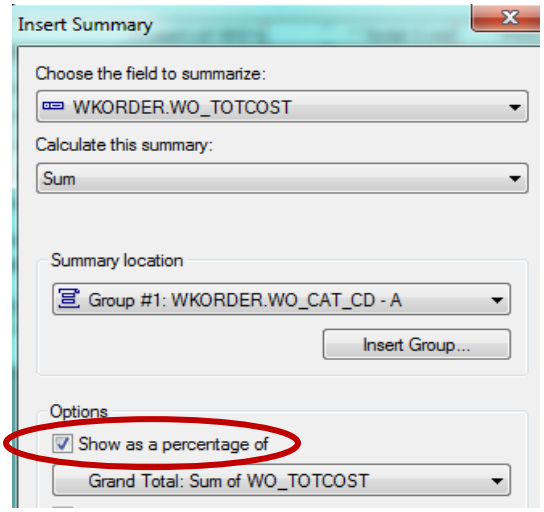
It is possible to show a value as a percent of a group or of the total using Insert Summary or a formula. These options may be used with reports that will **not** be run with filters on grid (child) fields.

Percentage with Insert Summary

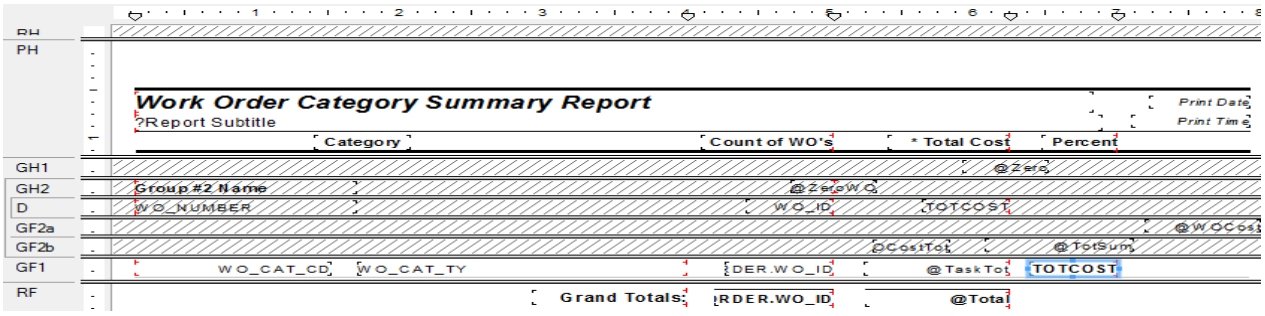
In the **Work** module there is a **Work Order Category Summary Report** (**WOCatSum.rpt**) that we will find the percent of the total cost of all work orders for that particular category.

1. From the Work report folder open **LC_WOCatSumPerc.rpt.**
2. Modify the existing report to allow room for the new Percent column.
    a. Reduce the **CAT_TY** field width by moving the right edge to 4”.
    b. Move the **Grand Totals** text field so that the right edge is also at 4”.
    c. Move the **Count of WO’s** column so that the right edge is at 5”.
    d. Move the **Total Cost** column so that the right edge is at 6 ¼”.
3. Add a text object for the **Percent** column.
    - Right align at 7”.

4. Calculate the value for the Category Cost as a percent of the Total Cost.
    - Click on Insert Summary. Set up as follows and click *OK*.



5. Move the summary that has been created so that the right edge aligns with the right edge of the **Percent** column.



41000	Potable	44	$2,596.80	0.58 %
41110	Potable Mains	124	$20,068.40	4.47 %
41120	Potable Nodes	1	$59.58	0.01 %
41130	Potable System Valves	73	$18,723.59	4.17 %
41150	Potable Vaults	4	$607.95	0.14 %
41200	Hydrants	421	$41,369.28	9.21 %

Percentage with a Formula

Continuing with the previous report, we will add the same percent using a formula.

1. Right Click on *Formula Fields* and select *New*.
2. Type in a name (**CatPercent**).
3. Expand *Functions* > *Summary* > *PercentOfSum*
4. Double click *PercentOfSum (fld, condFld)*

> *From the Crystal Report Help Guide:*
>
> *fld is a Number or Currency field that can be evaluated by the function.*
>
> *condFld is a field used to group the values in fld by.*

- The following formula shows in the Workshop window.

  PercentOfSum (,)

5. Expand *Report Fields*.
6. The *fld* to be used is the **WO_TOTCOST** field.
   - Click just to the right of the left parenthesis.
   - Double click the **WO_TOTCOST** field.
7. The *condfld* group field is **WO_CAT_CD**.
   - Click just to the right of the coma.
   - Double click **WO_CAT_CD**.
8. Click *Save and close*.
9. Drag the formula (**CatPercent**) just to the right of the Percent formula created with the *Insert Sum* tool.
10. While the new formula is highlighted, click the percent sign (%) in the tool bar.

This newly created formula should have the same percent value as the one created with the Insert Summary Tool.

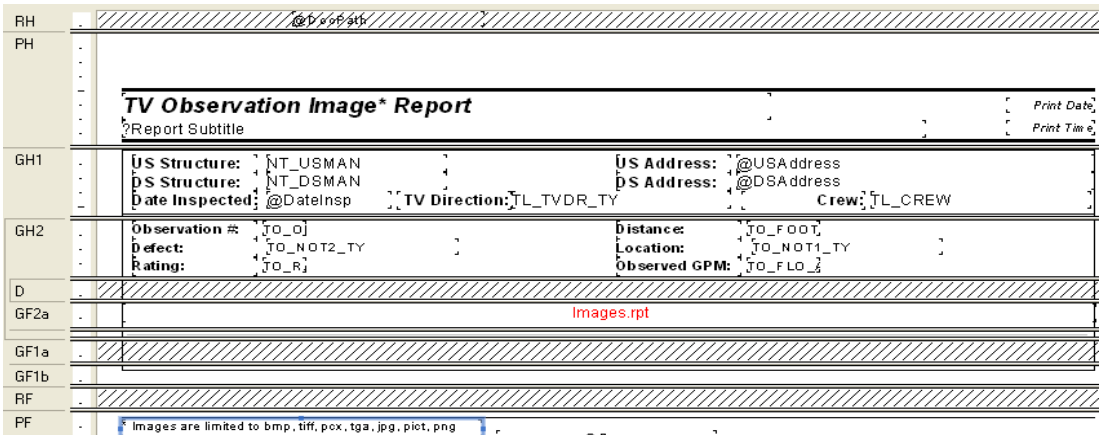
Example 29A

Image Reports

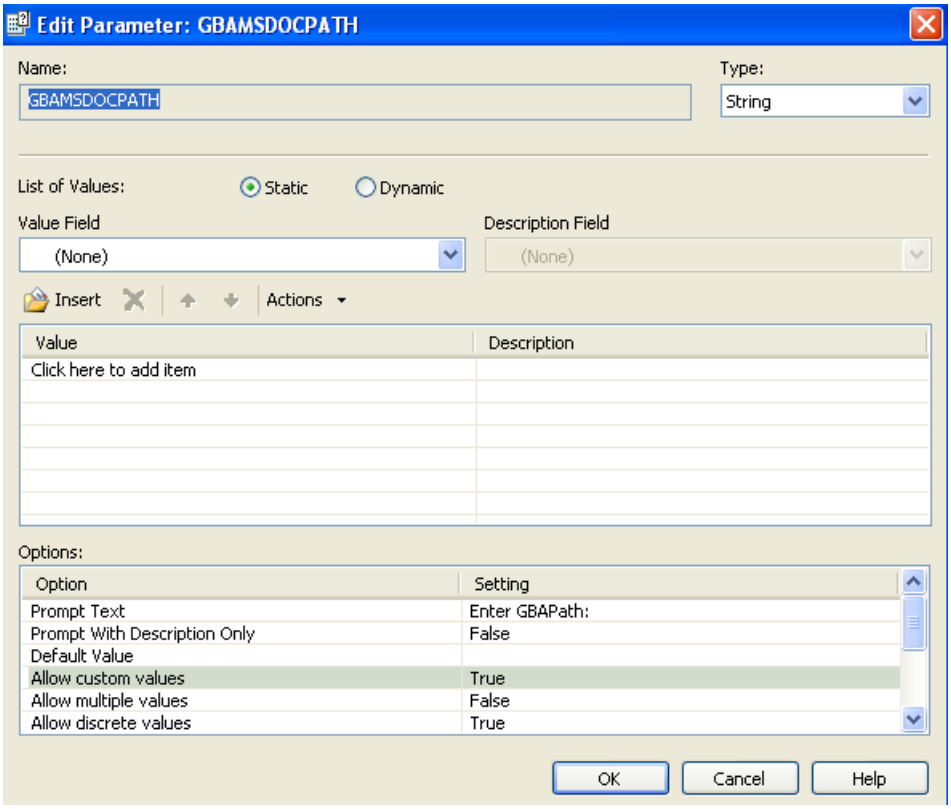
Reports using the attached images in Document control can be helpful. There are several imaging reports used throughout the Lucity modules. They support the following image types: JPG, BMP, TIF and PNG.

Desktop

The following items are necessary to bring an image into a report. We will look at the **TV Observation Image Report** (**TVObsrImg.rpt**).



1. First a *Parameter field* is set up and must have the Name of **GBAMSDOCPATH**:

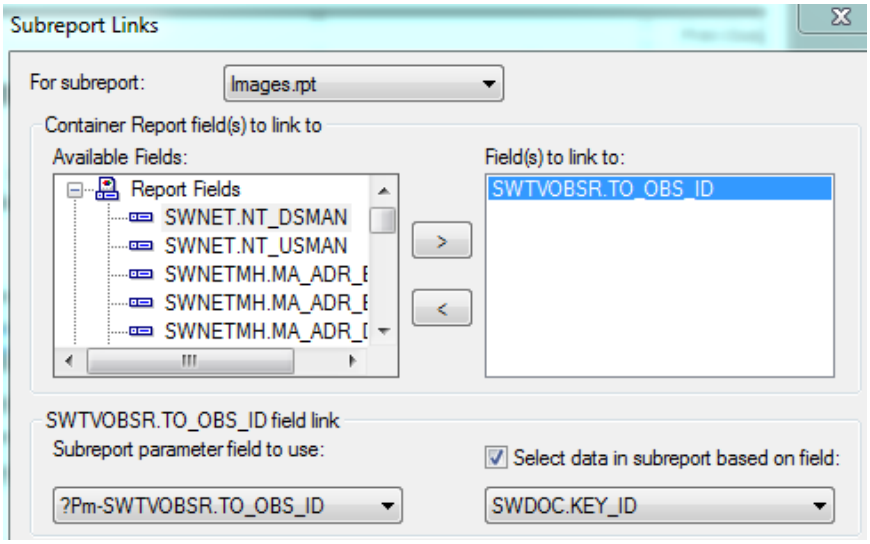


2. Then the **DocPath** formula is created and placed in the *Report Header*:

   shared stringvar GBADocPath:={?GBAMSDOCPATH};

3. The Image subreport is in a *Group Footer*. This group being on the ID field of the record the images are associated with. In this case, SWTVOBSR.TO_OBS_ID.

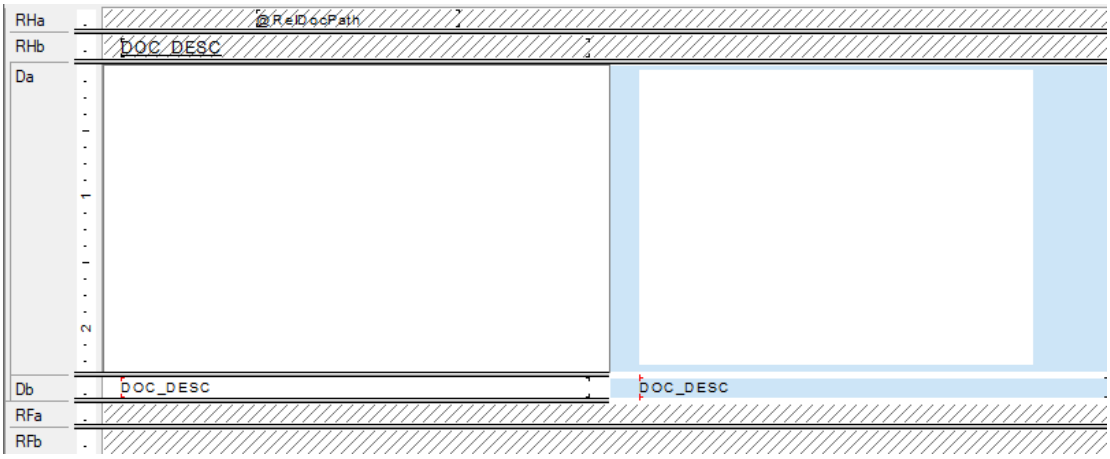
4. The subreport is linked to the main report:



5. Within the subreport's *Select Expert* is the following selection criteria:

   {SWDOC.DOC_PATH} Like ['*.jpg', '*.bmp', '*.jpeg', '*.tif', '*.tiff', '*.TGA', '*.PNG','*.JPG', '*.BMP', '*.JPEG', '*.TIF', '*.TIFF']
   and
   {SWDOC.KEY_ID} = {?Pm-SWTVOBSR.TO_OBS_ID} and
   {SWDOC.CHILD_TABLE} = "SWTVOBSR"

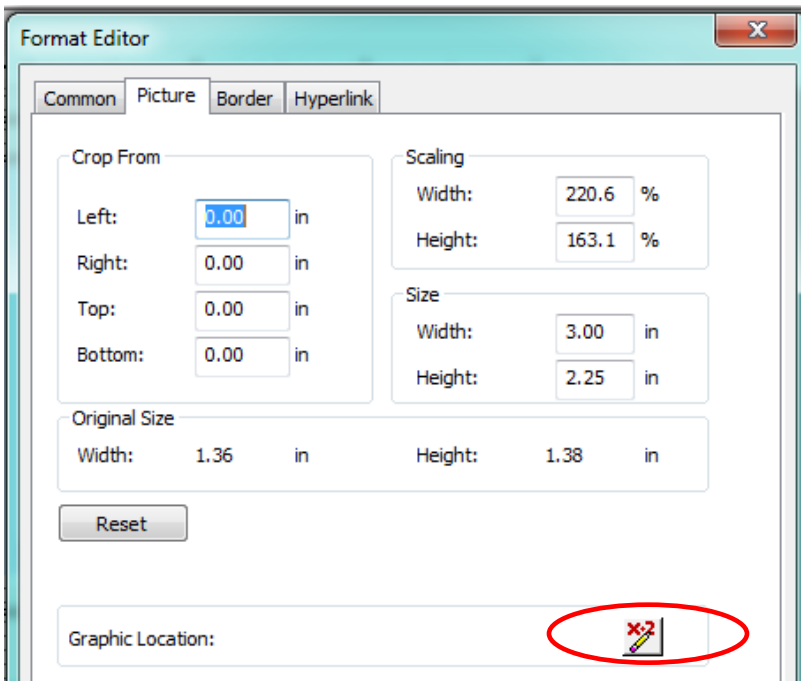
Note: Some Oracle versions are case-sensitive. The image may not show if the Path extension case does not match exactly.



6. The **RelDocPath** formula in the subreport *Header* looks like this:

   shared Stringvar GBADocPath;
   GBADocPath

7. The image is in an OLE object in the *Detail* section. Right click the object, select *Format Graphic*, and then click on the *Picture* tab.



Bringing the actual image in has been pretty tricky. Many reports have the OLE object set as above with the "Can Gow" option unchecked in the Common tab. This has resulted in some image distortion.

Better results have been achieved with actually adding a photo and scaling to 10% then checking the "Can Grow" option in the Common tab. The graphic Location still needs to be set as described in the next step.

This tends to be a trial and error process with your system's actual images.

8. Click the formula button next to Graphic Location. The following formula appears:

   shared stringvar GBA DocPath;
   if left ({SWDOC.DOC_PATH},13) = "$GBAMSDOCPATH"
   then {@RelDocPath} + right ({SWDOC.DOC_PATH}, (Len ({SWDOC.DOC_PATH})-13))
   Else {SWDOC.DOC_PATH}

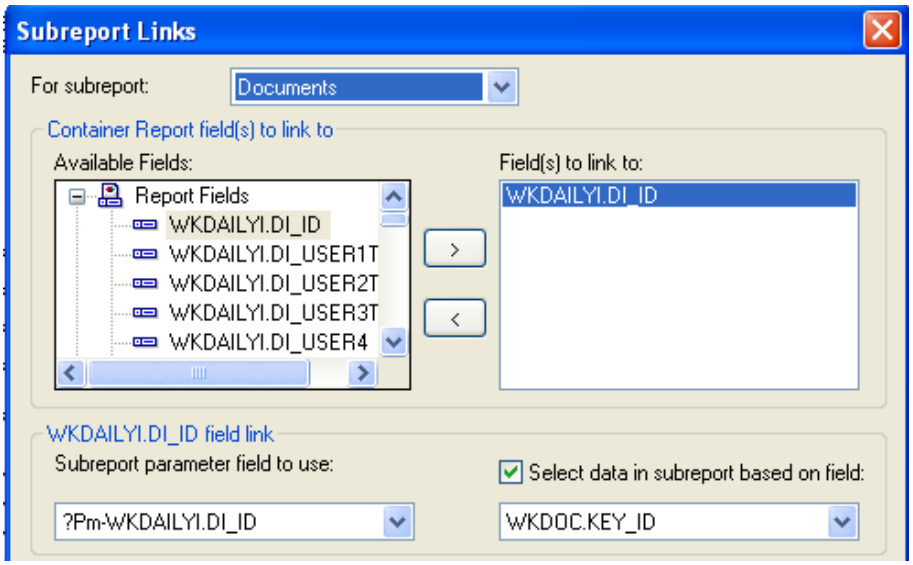
9. To suppress the *Detail* section when there is no image:
   a. In *Section Expert* click on the *Details a* section.
   b. In the Suppress (No Drill-Down) formula box there is a formula.

      isnull ({SWDOC.DOC_ID})

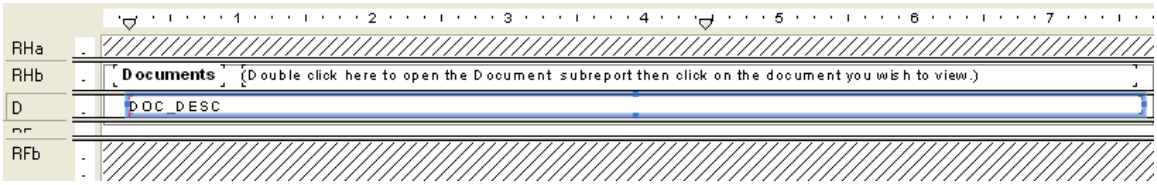
Hyperlinking

Modules can have documents attached to them. A client wished to have a link to these documents in the **Daily Inspection Detail Report**.

A *Documents* subreport was added. The subreport used the **WKDOC** table. It was linked as follows:



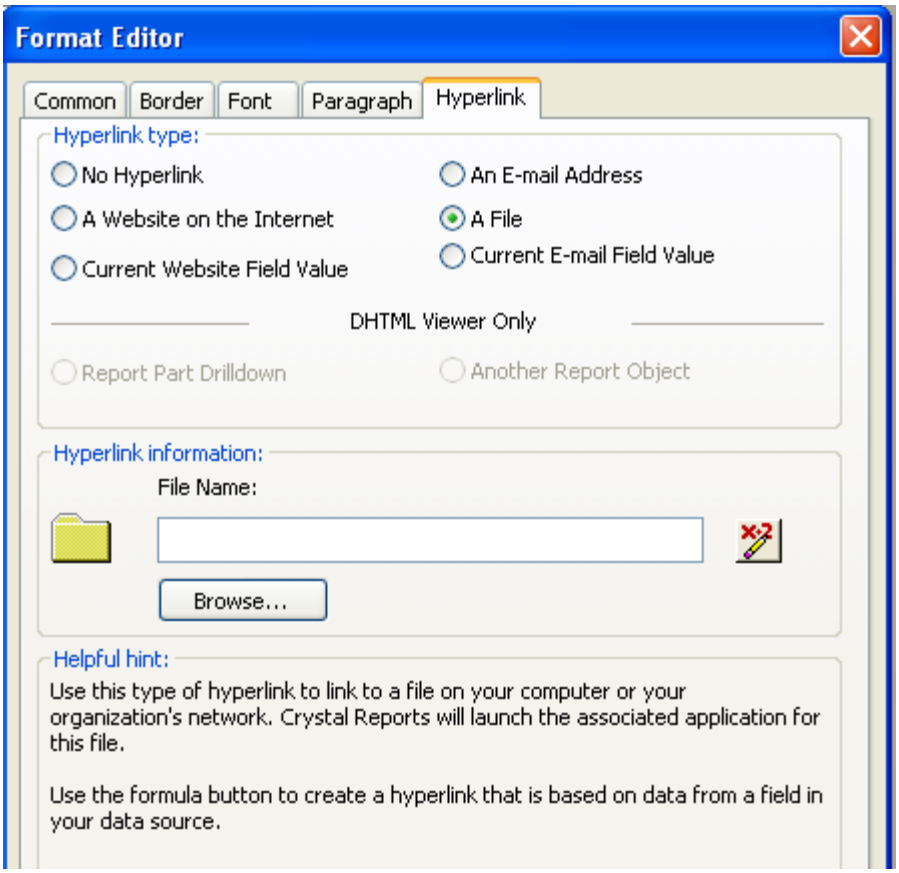
The subreport was set up like this:



- Within the *Select Expert* the formula looked like this:

  {WKDOC.MODULE_NAME}="WKDAILYI" and
  {WKDOC.KEY_ID} = {?Pm-WKDAILYI.DI_ID}

- For the **DOC_DESC** field you right clicked and selected *Format Field*.
- Then the *Hyperlink* tab was selected.



- In the ***File Name*** *Formula Workshop* the Document Path field was brought in:

  {WKDOC.DOC_PATH}

- In the subreport *Report Header* a statement was added so the report user knew what to do to access the documents.

  - (Double click here to open the Document subreport then click on the document you wish to view.)

Use of Commands

If the database you are using supports SQL, you can write your own command which will be represented in Crystal Reports as a Table object. This can reduce the size of the data being brought in as well as running fields through calculations and groupings before the data runs in the report. Sorting can be an issue when the value to be sorted on is a formula within a group. The *Group Sort Expert* can be helpful but not all formula fields are available for sorting.

Commands do not recognize Lucity filters when the report is run.

Sort on Calculated Field

Here is an example of a report which only used a Command.

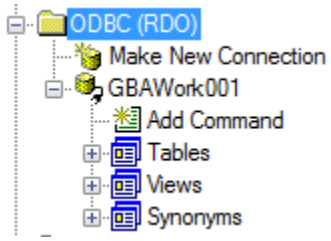
Task Summary Report - Avg Cost/Unit Sort

7/17/2014
9:57:54PM

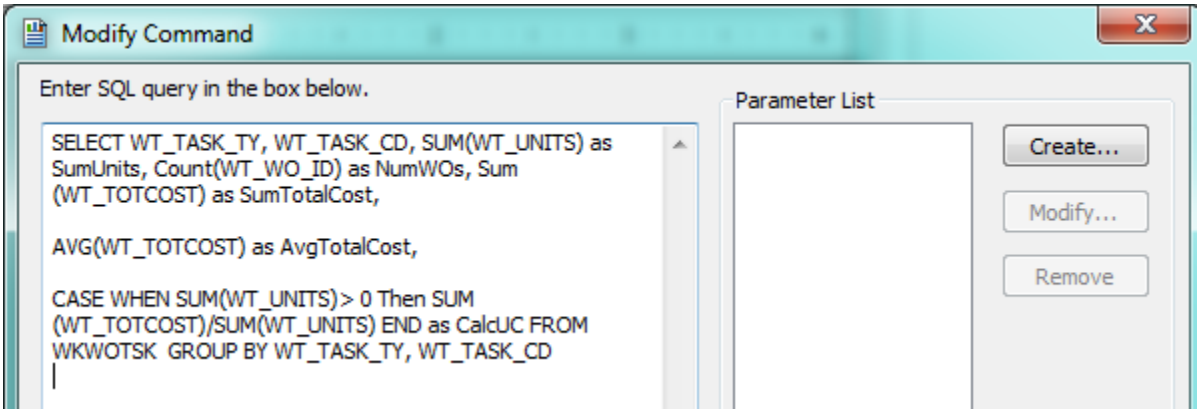
Task	**# WOs	Units	*Total Cost	*Avg Cost/Unit
ENWT00 Engineering Quality Control	103	3.00	$114,320.94	$38,106.98
VMFUEL01 Semi-Annual Dispenser Maint	13	2.00	$14,743.95	$7,371.98
VMFAB01 Fabrication	62	54.00	$328,978.20	$6,092.19
CSINFO01 Info Calls Solid Waste	7,710	720.00	$1,927,740.94	$2,677.42

The report needed to group on the Task but the sort was on the **Avg Cost/Unit** formula field. This formula was not available in the *Group Sort Expert* so a Command was used.

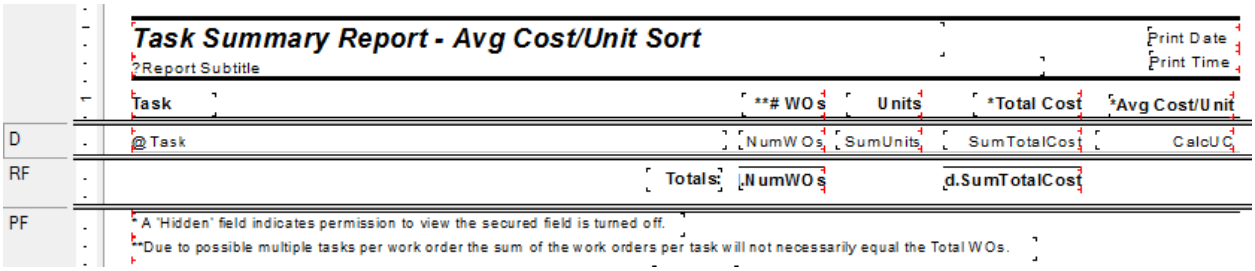
When the report was created and the Data window appeared, instead of selecting tables, the *Add Command* was selected.



The SQL statement was added to the left box.



The values were brought into the report and then the *Record Sort Expert* was used to sort on the **CalcUC** field.



Record Selection with "Or" Statement Using Multiple Tables

Another case where Add Command is used is when you need to Select records with an OR statement that uses fields from different tables.

To set this up you need to know the required tables, the fields used in the report and the selection criteria.

Here is an example of the SQL statement used to create this type of report:

SELECT SWBLDG.*, SWBLDGOB.*, SWBLDGSUMP.BB_STAT_TY, SWNET.NT_BASIN, SWNET.NT_USMAN, SWNET.NT_DSMAN, SWBLDGOB.BO_STAT_TY, SWBLDGSUMP.BB_STAT_TY

FROM ((SWBLDG LEFT JOIN SWBLDGOB ON SWBLDG.BD_BLDGID = SWBLDGOB.BO_BLDGID) LEFT JOIN SWBLDGSUMP ON SWBLDG.BD_BLDGID = SWBLDGSUMP.BB_BLDGID) LEFT JOIN SWNET ON SWBLDG.BD_NETID = SWNET.NT_ID

WHERE

SWBLDGOB.BO_STAT_TY='positive' or

SWBLDGOB.BO_STAT_TY='suspect' or

SWBLDGSUMP.BB_STAT_TY='positive' or

SWBLDGSUMP.BB_STAT_TY='suspect'

Limiting Parameter Choices in Dynamic Selection

A dynamic parameter offers a list of all data in a particular field. A Command may be used to limit the data being offered in the pick list.

The following command was created to limit the Wor Order Tasks to be selected from to those used for Street Signals.

SELECT WO_ACTN_TY FROM WKORDER WHERE WO_INV_ID=246

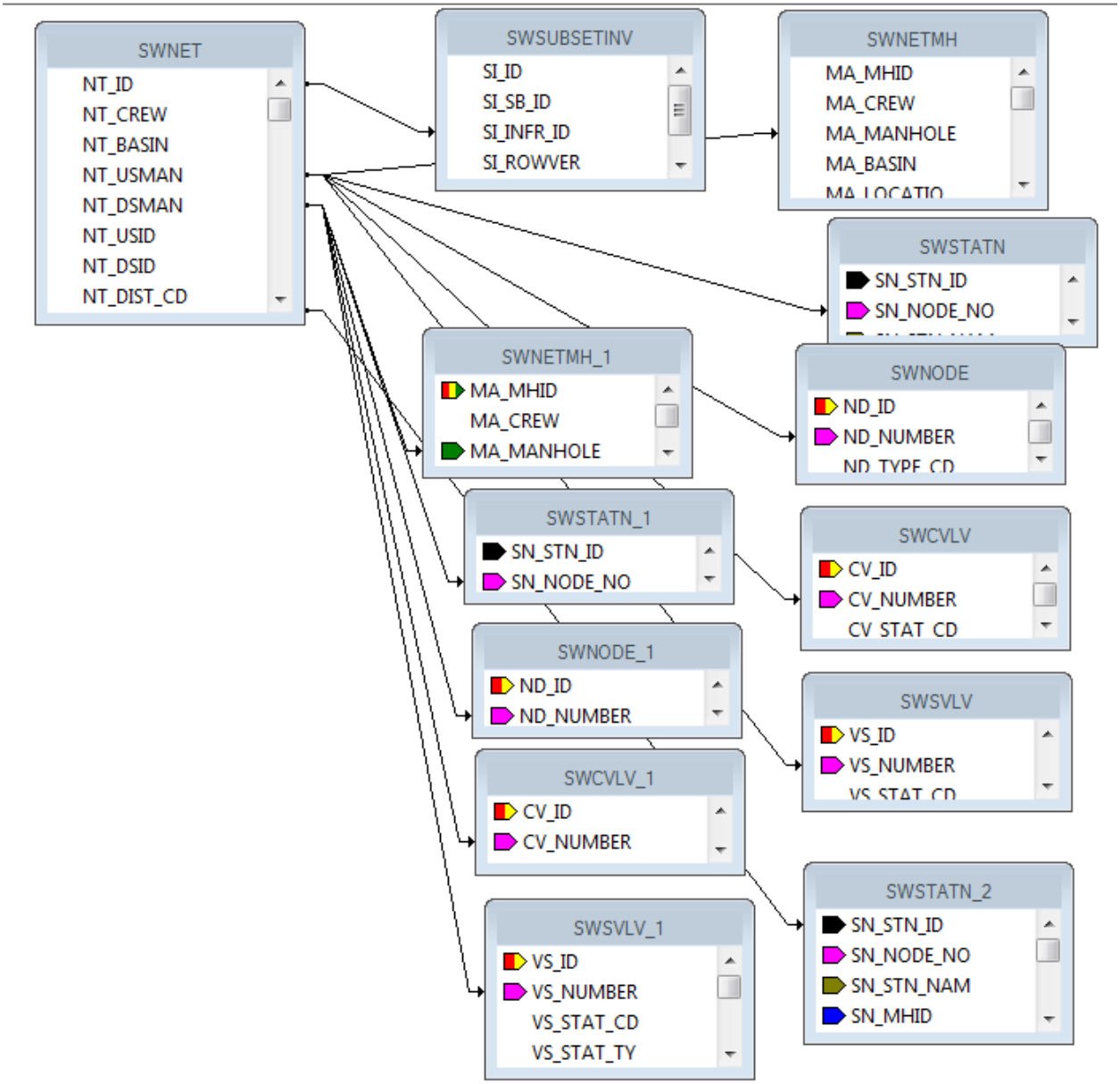
For Web reports the report will not offer the picklist on the first run. The parameter would need the editable option and the report run with the Advanced View so the report could be run a second time using the parameter tab.

Duplicate Table Use

Due to Lucity's flexibility in the use and linking of tables it is helpful that Crystal can accommodate most of the relationships.

Crystal Reports allows the same table to be brought in multiple times into a report by adding a "_#" such as "_1" or "_2" to the end of the table name.

This is helpful when showing the relationship of the structures to the ends of a pipe as seen in the **Sewer Pipe Detail Report** (**PipeDet.rpt**).



Conversions

The fields in Lucity can be a variety of types like numeric, text or datetime. If the fields are used in a formula, then all of the pieces need to be the same type of data. You can use conversion formulas:

ToText(x)

ToString(x)

ToNumber(x) - There could possibly be issues with this formula. An optional conversion for a string filed is VAL(x)

To go from all caps to "Normal Text" use the following formula:

ProperCase({WKRESRCE.WR_RSRC_TY})

To get years to show up without a comma:

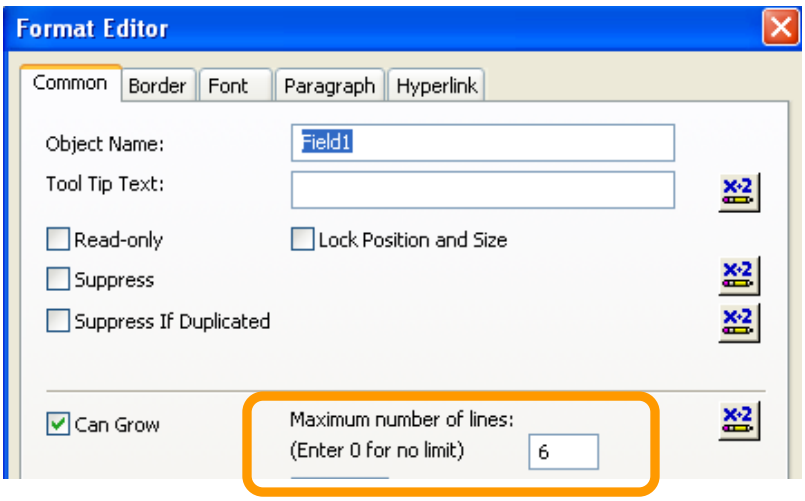
ToText(Year ({?Post Date}),0,'')

Odd Things Happen

Field not showing all of the text

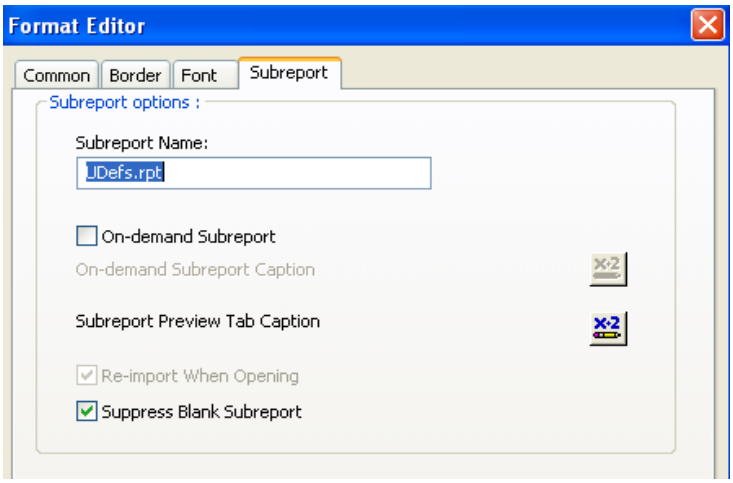
Check the Field Format Editor in the common tab to make sure the *Can Grow* option is checked. If it is checked and the field is still cutting off the data then make sure the maximum lines is set at "0".

Work – old version (**RTDetail.rpt**) formatting for the Comments For Crew subreport, **CO_TEXT** field.



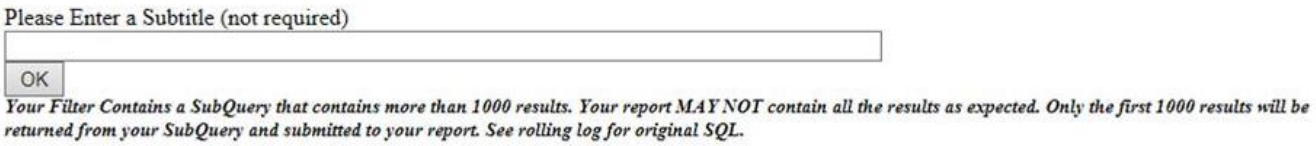
Report Takes a Long Time to Open

If you have a Detail report that takes a LONG time to open, check the **UDefs.rpt** subreport at the top of the report and the Comment subreport at the bottom of the report. Open the Format Editor dialog box for the Subreport and click on the Subreport tab. Make sure the "*Re-import When Opening*" check box is NOT checked. Sometimes this option is silver and can't be changed.



Filter Warning

Warning on filter results.



If there is an "and" statement in the filter, make sure to put the more restrictive part first in the statement to bring the record count down before the filter uses the second part of the statement.

This filter worked:

WKORDER WHERE WKORDER.WO_ID IN (SELECT WKGDMEMO.GM_PAR_ID FROM WKGDMEMO WHERE (WKGDMEMO.GM_PARENT ='WKORDER' and (WKGDMEMO.GM_Rec_DT Between #05/09/2016# AND #05/10/2016# AND WKGDMEMO.GM_MEMO LIKE '%ROOT%'))

The date portion limited the records to a reasonable number before the "Like" statement selected from the records.

Filter that did not work:

WKORDER WHERE WKORDER.WO_ID IN (SELECT WKGDMEMO.GM_PAR_ID FROM WKGDMEMO WHERE (WKGDMEMO.GM_PARENT ='WKORDER' and (WKGDMEMO.GM_MEMO LIKE '%ROOT%' AND WKGDMEMO.GM_Rec_DT Between #05/09/2016# AND #05/10/2016#))