TRAINING GUIDE

Advanced Crystal 2

clarity through connections www.lucity.com 10561 Barkley, Suite 500 Overland Park, KS 66212 913.341.3105

Using Crystal Reports with Lucity

Advanced Examples - 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. In the following step-by-step example, we'll 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.

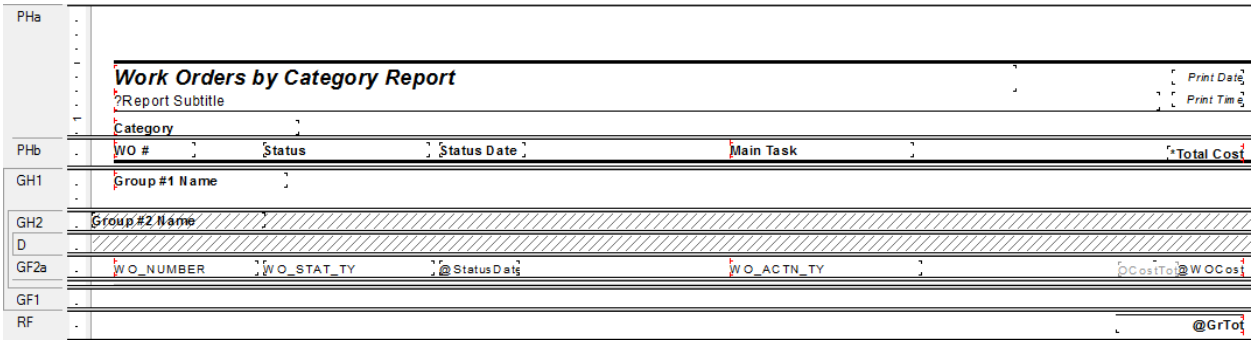
Category	Count
13310 Auxiliary Equipment	1
41310 Backflow Preventors	1
12100 Bridge Maintenance	1
12510 Bus Stops	1
13320 Cabinet	1
02000 Call Center	3
92100 Commercial Collecti	2
13341 Conductor Cabling	1
13342 Conduit Cabling	1
12520 Curb	1
11300 Detention Basins	1
60000 Equipment	99

The hidden section of the Equipment category looks like this:

Equipment

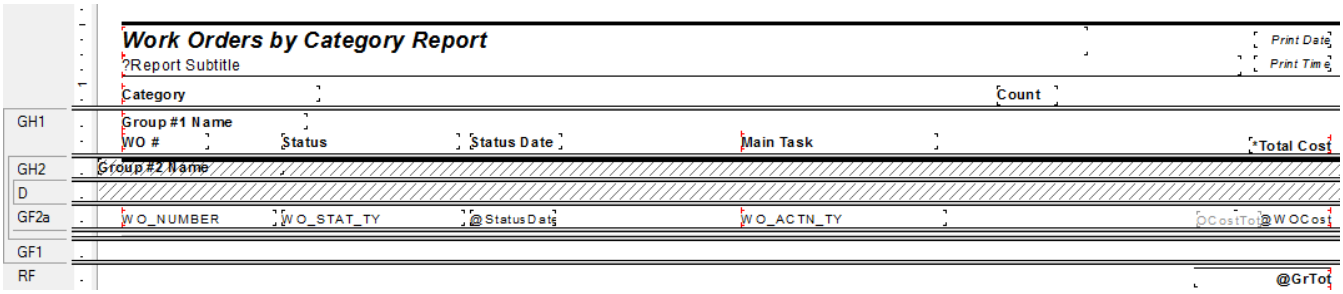
WO #	Status	Status Date	Category	Main Task	*Total Cost
2006-01662	Complete	7/25/2006	Equipment	Miscellaneous	$156.59
2006-01682	Complete	8/7/2006	Equipment	Dist Maintenance/Equipment	$155.47
2006-01684	Complete	8/7/2006	Equipment	Miscellaneous	$208.35
2006-01685	Complete	8/7/2006	Equipment	Miscellaneous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

1. From the Work Order module, export the **Work Order Summary Report** (**WOSum.rpt**) and rename it **LC_WOHide.rpt**.
2. Delete the **Address** column.
3. Add the **Category** grouping and move the Category grouping to be the first grouping.
4. 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.
5. Move the **Category** title above **WO #** and remove the Category field.
6. Increase the *Group Header 1* section size.

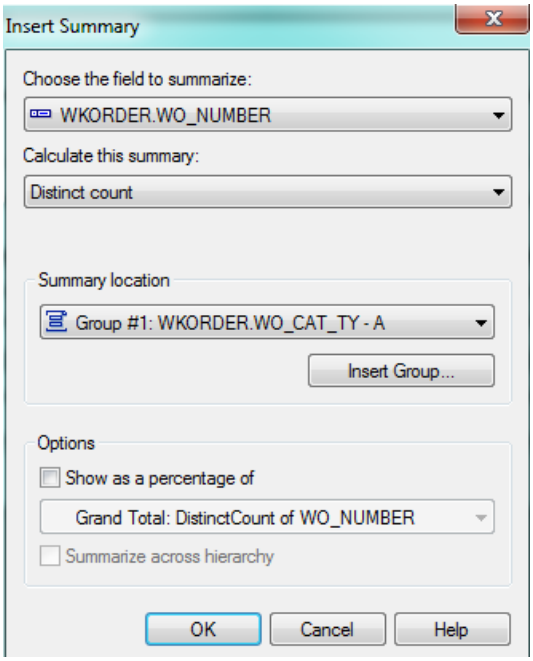


7. Right click in the *Page Header b* section, click *Select All Section Objects*.
8. Drag the column titles into the *Group Header 1* section, beneath **Group #1 Name**.

9. Right click in the *Page Header a* section and select *Merge Section Below*.
10. Add a column title called **Count** in the *Page Header* section.



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

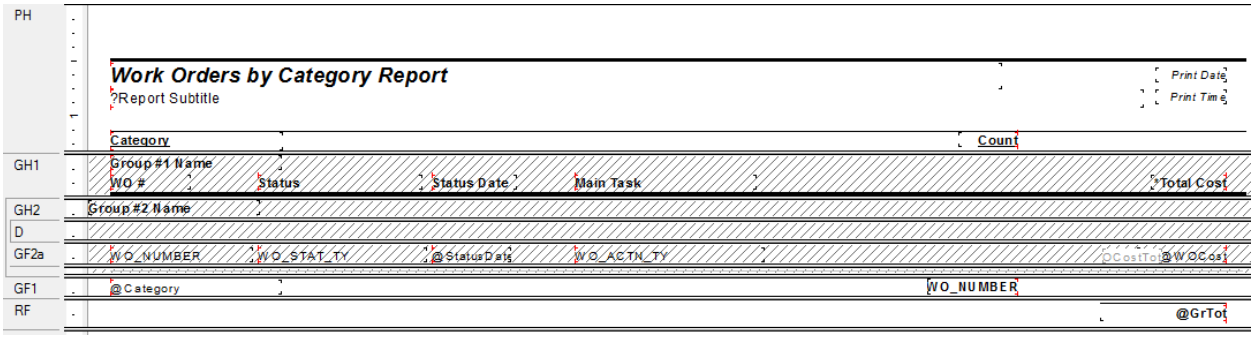


12. Next, we will create a concatenated formula where fields can be "tacked together" to eliminate awkward spacing. Create a new formula called **Category**.

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

13. Drag this formula into the *Group Footer 1* section under the **Category** column title.
14. To the left of the report, right click in the *Group Header 1* section and select *Hide(Drill-Down)*.

15. Repeat step 14 for the *Group Footer 2a* section.



16. 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

17. 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

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.

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

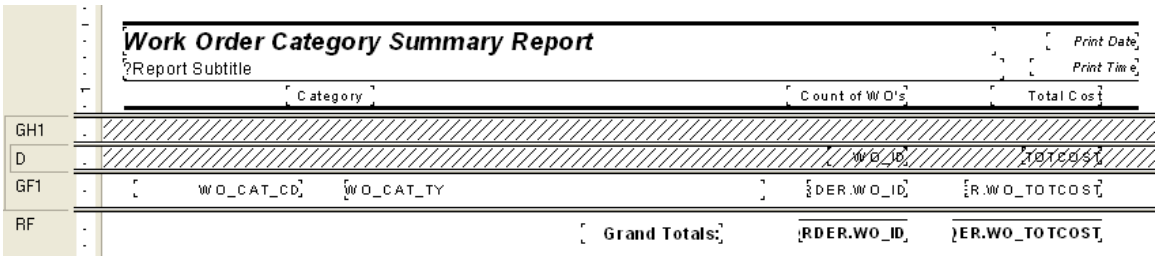
Evaluation Times are statements that tell when the formula is to be evaluated. If nothing is stated, Crystal Reports guesses what is appropriate for the data being used in the formula.

- **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

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:

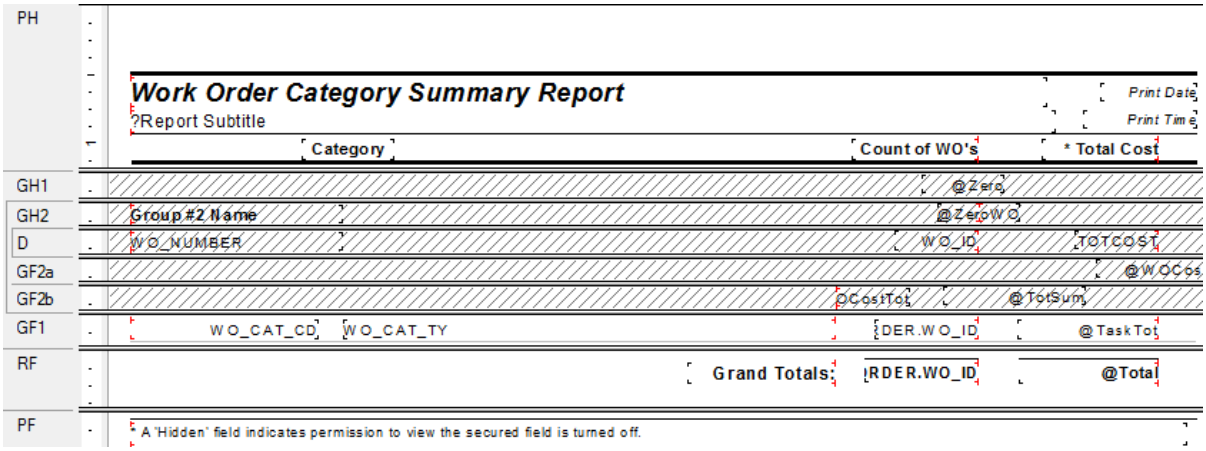
3010	Tree	2	$2,143.54
40000	Water Department	25	$464.08
41000	Potable	40	$10,403.04
41110	Potable Mains	122	$59,106.27

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

Open **WOCatSum.rpt** and follow the steps used to correct the report.



1. Added a second grouping on the Work Order Number, **WO_NUMBER**.
2. Created a formula for the Work Order Cost (**WOCost**) and placed it in the WO Number *Group Footer 2*.

   WhilePrintingRecords;

   Shared numberVar WOCost ;

   WOCost:={WKORDER.WO_TOTCOST}

3. Created a formula to summarize the cost for the Category (**WOCostTot**). Added 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 WOCostTot ;

   Shared numberVar WOCost ;

   WOCostTot:= WOCostTot + WOCost

4. Created a formula to summarize the total cost for the Work Orders (**TotSum**). This was placed in *GF2b*.

   WhilePrintingRecords;

   Shared numberVar GrWOCost ;

   Shared numberVar WOCost ;

   GrWOCost:= GrWOCost + WOCost

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

   Shared numberVar WOCost :=0;

6. Created a formula to reset the Category Cost variables (**Zero**). Placed this in the Category *Group Header #1*.

   Shared numberVar WOCostTot :=0 ;

   Shared numberVar WOCost :=0;

7. Created a formula to show the Category total cost (**TaskTot**). Placed this in the Category *Group Footer #1*.

   WhilePrintingRecords;

   Shared numberVar WOCostTot ;

   WOCostTot

8. Created a formula to show the Grand total cost (**Total**). Placed this in the *Report Footer*.

   WhilePrintingRecords;

   Shared numberVar GrWOCost ;

   GrWOCost

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

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. As seen in Unlinked Subreports on the next page.

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:

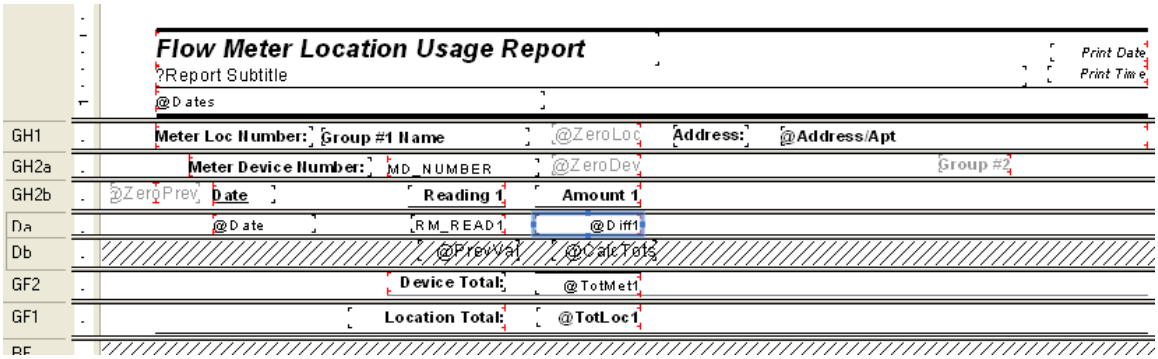
WhilePrintingRecords;

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 (**ZeroLoc**):

  shared Numbervar LocTot1:=0;

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

  shared Numbervar MetTot1:=0;

- In *GH2b* there is a formula to reset (**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

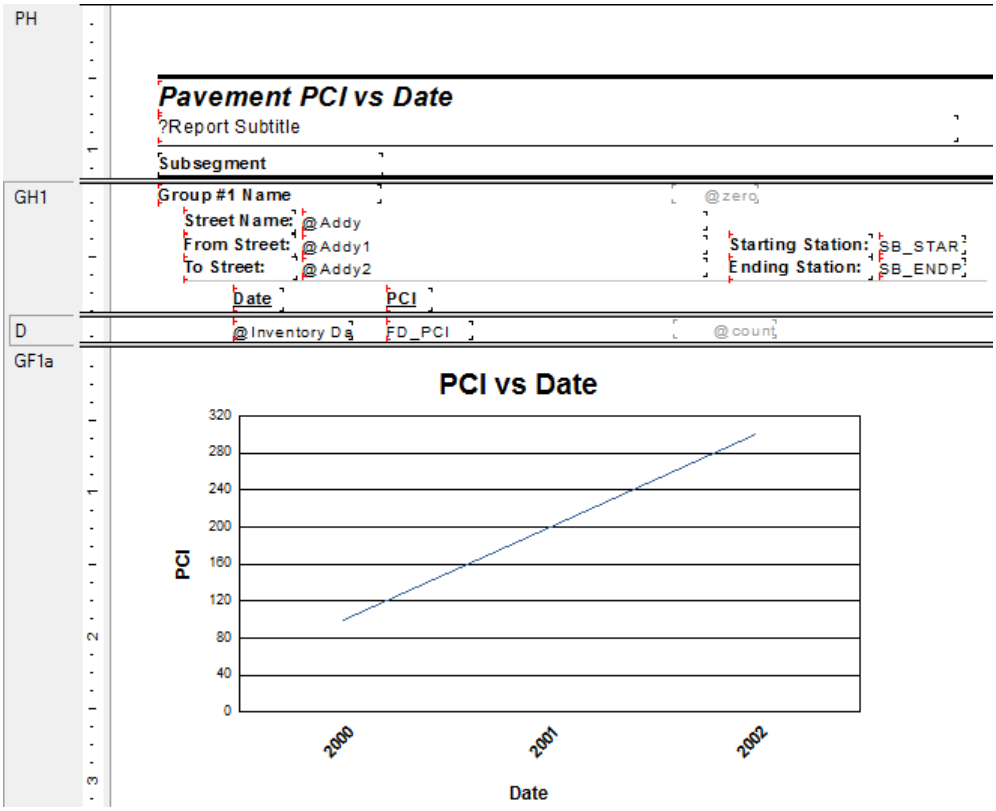
- In *GF1* there is a formula for the Location total usage (**TotLoc1**):

  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.

Unlinked Subreports

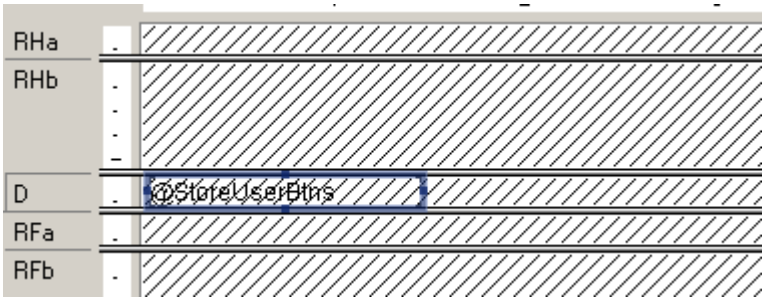
There are times when the subreports do not need to be linked. For example, you won't need to link subreports if there are no connections to the main report. In this situation, the subreport can share data using variables but does not necessarily share a database.

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 there are two ways to alter Field captions to suit individual needs:

- 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 detailed reports. It is hidden in a Report Header subreport. 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
```

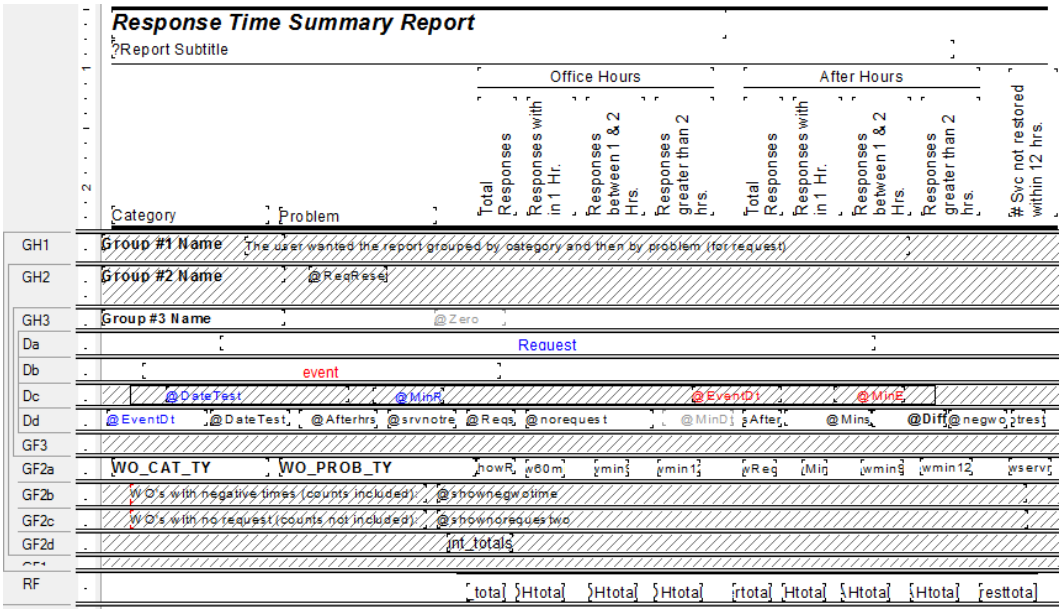
```
//The subreport that this formula belongs to must contain the following
//Select Expert statement:
//{WTFIELDS.FieldName} like ["BD_USE*"]
//The Fields table reference should match your database.  The two character
//data field prefix and suffix must be changed for each report.
//You must also enter 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}+':';
```

Using Variables and Documentation

A customer report was sent in to be modified to use dates and times from the Work Order Event grid and the Request grid. This was the best documented report I have ever seen. This example is more to show the extent that variables can be used and the documentation.

Example portion (all lines starting with // are strictly for documentation):

```
// Here we are determining what time category the time difference falls into
// We declare the variable for use in the formula.
// It is shared because we are passing the value to and from other formulas.
// These are variable that we cleared in group 2 header.
    shared numbervar Min60_OH;
    shared numbervar Min90_OH;
    shared numbervar Min120_OH;
    shared numbervar Min60_AH;
    shared numbervar Min90_AH;
    shared numbervar Min120_AH;
//shared stringvar negwotime;
    if ((not ({@DateTest} = ''))and (not({@EventDt}=''))) then
// we take the current value of the variable
// and make it equal to that value plus the current value of the @Mins formula.
    if {@Mins} <= 60 and {@Mins} >= 0 and {@Afterhrs} = 0 and not (totext ({@DateTest}) = '') then
    Min60_OH:= Min60_OH + 1;
    if {@Mins} > 60 and {@Mins} <= 120 and {@Afterhrs} = 0 and not (totext ({@DateTest}) = '') then
    Min90_OH:= Min90_OH + 1;
    if {@Mins} > 120 and {@Afterhrs} = 0 and not (totext ({@DateTest}) = '') then Min120_OH:=
    Min120_OH + 1;
    if {@Mins} <= 60 and {@Mins} > 0 and {@Afterhrs} <> 0 and not (totext ({@DateTest}) = '') then
    Min60_AH:= Min60_AH + 1;
    if {@Mins} > 60 and {@Mins} <= 120 and {@Afterhrs} <> 0 and not (totext ({@DateTest}) = '')
    then Min90_AH:= Min90_AH + 1;
    if {@Mins} > 120 and {@Afterhrs} <> 0 and not (totext ({@DateTest}) = '') then Min120_AH:=
    Min120_AH + 1;
    if {@Mins} < 0 then negwotime = negwotime & " " & {WKORDER.WO_NUMBER} & ",";
```



The Request Date and Time came from the Request subreport and the Event Date and Time came from the Event subreport. These values were brought to the main report as variables.

There were a number of formulas using variables so I really didn't want to start from scratch. Therefore, I worked with what was there. This was the reason for some of the challenges.

From the way the date field was originally used, it needed to be a text field.

The date and time fields were then handled separately. The difference in days were taken and then the difference in times.

The days were converted to minutes:

// In this step, we want to display the time difference from when the request was entered

// to when the first task was started.

// First we see if there are values in the date formulas (the Request and Event dates)

```
shared Numbervar DtMin;
if ((not ({@DateTest} = ''))and (not({@EventDt}=''))) then
```

// If there are values, then we get the difference between the two date values.

```
DtMin:= DateDiff ("n", Date ({@DateTest}), Date ({@EventDt}));
```

// the "n" part indicates that we want minutes returned.

The time portions were then looked at.

The Request minutes were found:

```
RMin:=60*(hour(time({WKREQ.RQ_REC_TM}))) + minute(time({WKREQ.RQ_REC_TM}))
```

And then the Event minutes were found:

```
EMin:=60*(hour(time({WKWOEVENTS.WE_EVNT_TM}))) +
minute(time({WKWOEVENTS.WE_EVNT_TM}))
```

Then the Difference in Date/Time of the Request being entered and the Event occurring was found (@Mins):

```
if ((not ({@DateTest} = ''))and (not({@EventDt}=''))) then  DtMin - Rmin + Emin
```

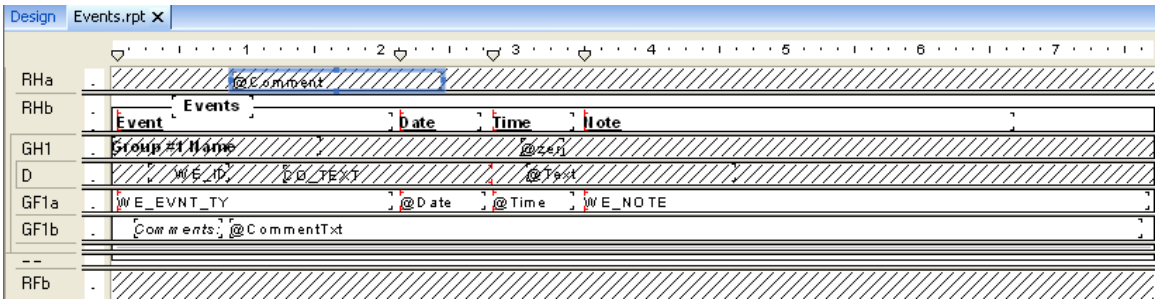
Comments to Non Comment Subreports

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.

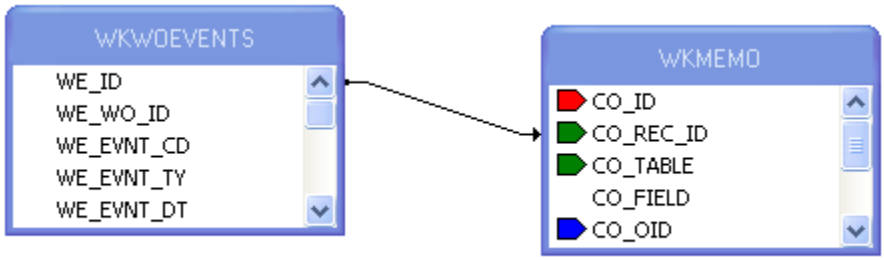
There are a variety of ways to accomplish this task. Not all work for the various scenarios.

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.



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.

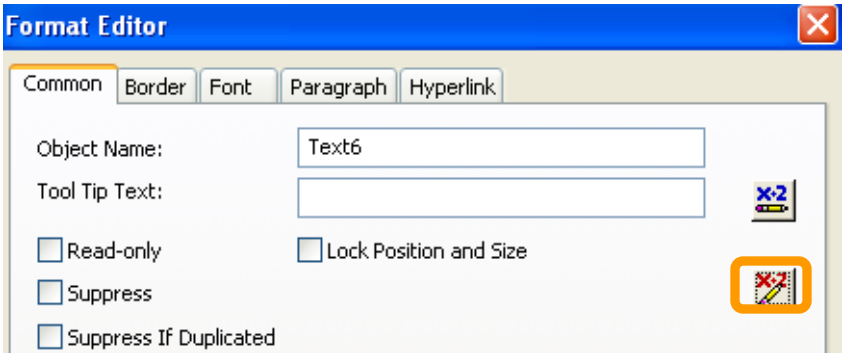
```
WhilePrintingRecords;
Shared stringVar Text:=" ";
```

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.



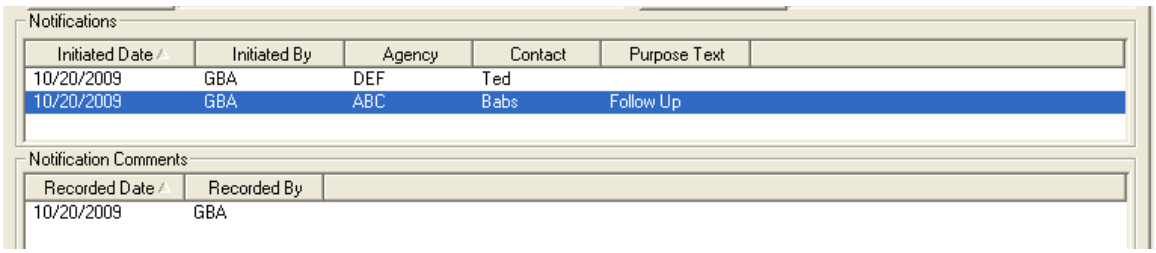

```
WhilePrintingRecords;
shared stringVar Text ;
Text=" "
```

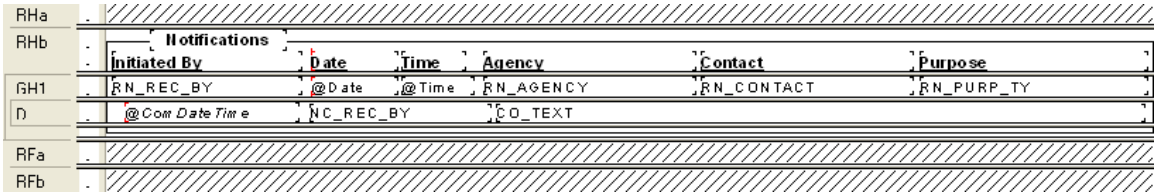
9. In Section Expert select *Suppress Blank Section* for the second *Group Footer*.

Option 2 - Grouping & Conditional Suppression

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

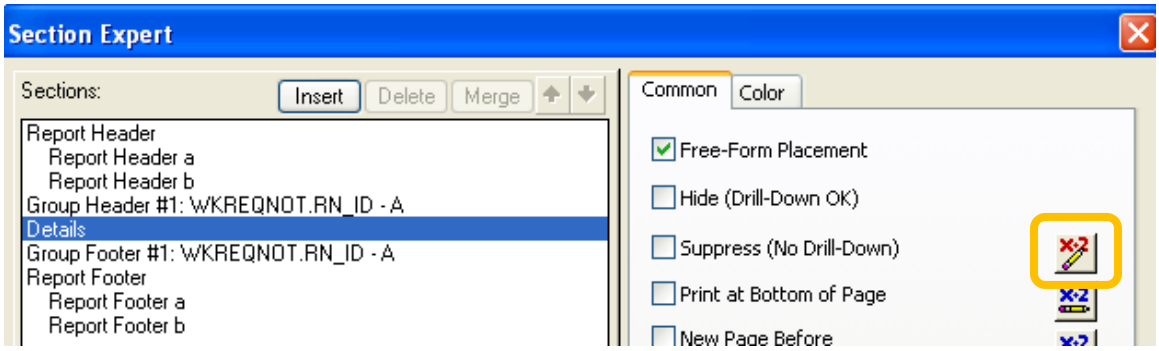
The comment was added to the *Notifications* subreport within the **Request Detail Report** (**ReqDetail.rpt**).





1. Bring in the **WKMEMO** table and link.

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"

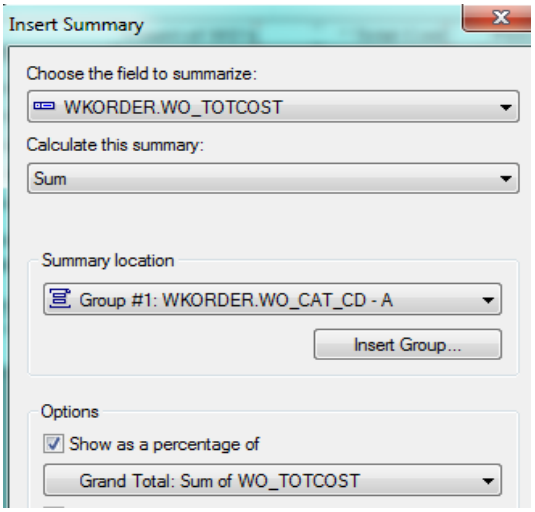
Percentages

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

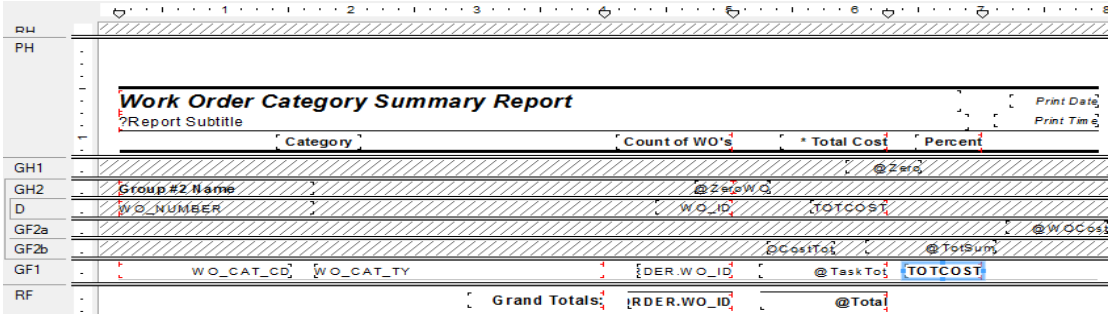
Percentage with Insert Summary

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. Use a copy of the **Work Order Category Summary Report** and name it **LC_WOCatSumPerc.rpt.**
2. Modify the existing report to allow room for the new Percent column.
    - Reduce the **CAT_TY** field width by moving the right edge to 4".
    - Move the **Grand Totals** text field so that the right edge is also at 4".
    - Move the **Count of WO's** column so that the right edge is at 5".
    - 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.

Percentage with Variables

The **Work Order Category Summary Report** currently uses variables to calculate the total cost for the Category as well as the Grand Total.

If you try to use the variables in a percent calculation – something like this:

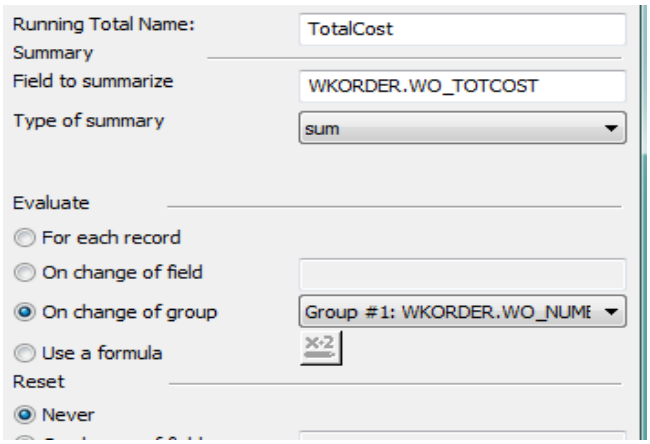
```
if {@Total}>0 then
100 * {@TaskTot}/{@Total}
```

The **@Total** value isn't for the entire report. **@Total** is a running total that will only be the total up to that point. This value is shown on the far right.

41000	Potable	44	$2,596.80	0.58 %	0.58 %	21.38 %
41110	Potable Mains	124	$20,068.40	4.47 %	4.47 %	62.30 %
41120	Potable Nodes	1	$59.58	0.01 %	0.01 %	0.18 %
41130	Potable System Valves	73	$18,723.59	4.17 %	4.17 %	36.71 %
41150	Potable Vaults	4	$607.95	0.14 %	0.14 %	1.18 %
41200	Hydrants	421	$41,369.28	9.21 %	9.21 %	44.50 %

A total value can be calculated in a subreport before it is used in the main report.

1. Click *Insert Subreport*.
2. Give it a name (**TotalCost**).
3. Click *Report Wizard*.
4. Bring the **WKORDER** table over to the *Selected Tables* box. *Next*
5. Bring the **WO_NUMBER** and **WO_TOTCOST** fields over to *Fields to Display*. *Next*
6. Bring **WO_NUMBER** field over to *Group By*. *Next*
7. Summararized Fields should be **WO_TOTCOST**. *Next*
8. *Group Sorting* should be **WO_NUMBER.** *Next*
9. No Chart. *Next*
10. No Record Selection. *Next*
11. No Template. *Finish*. *OK*
12. Drop the subreport into the *Report Header* and unsuppress the section.
13. Format the subreport.
14. Open the subreport.
15. We will now create a variable that totals all Work Order Costs.
    - Right click on *Running Total Fields*.
    - Click *New*
    - Give it a name (**TotalCost**).
    - Set up as follows:



- Right click on *Formula Fields*
- Click *New*
- Give it a name (**TotalCostV**)
- Enter a formula for a variable equaling the *Running Total* formula.

```
WhilePrintingRecords;
shared numbervar TotCst;
TotCst:={#TotalCost}
```

- Click *Save and close*
- Drag the formula into the *Report Footer*.
- Suppress all sections of the subreport.

16. In the main body of the report modify the **CatPercV** formula to calculate the Category cost percent using the newly created variable for total.
    - Right click on the **CatPercV** formula and select *Edit*.
    - Change the formula as follows:

```
shared numbervar TotCst;
if TotCst>0 then
100 * {@TaskTot}/TotCst
```

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

All three options to calculate the percent gave the same value and could have been used in this situation. Usually it is personal preference as to which option is used but sometimes the report requires a specific method.

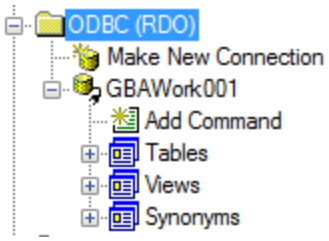
Use of Commands

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.

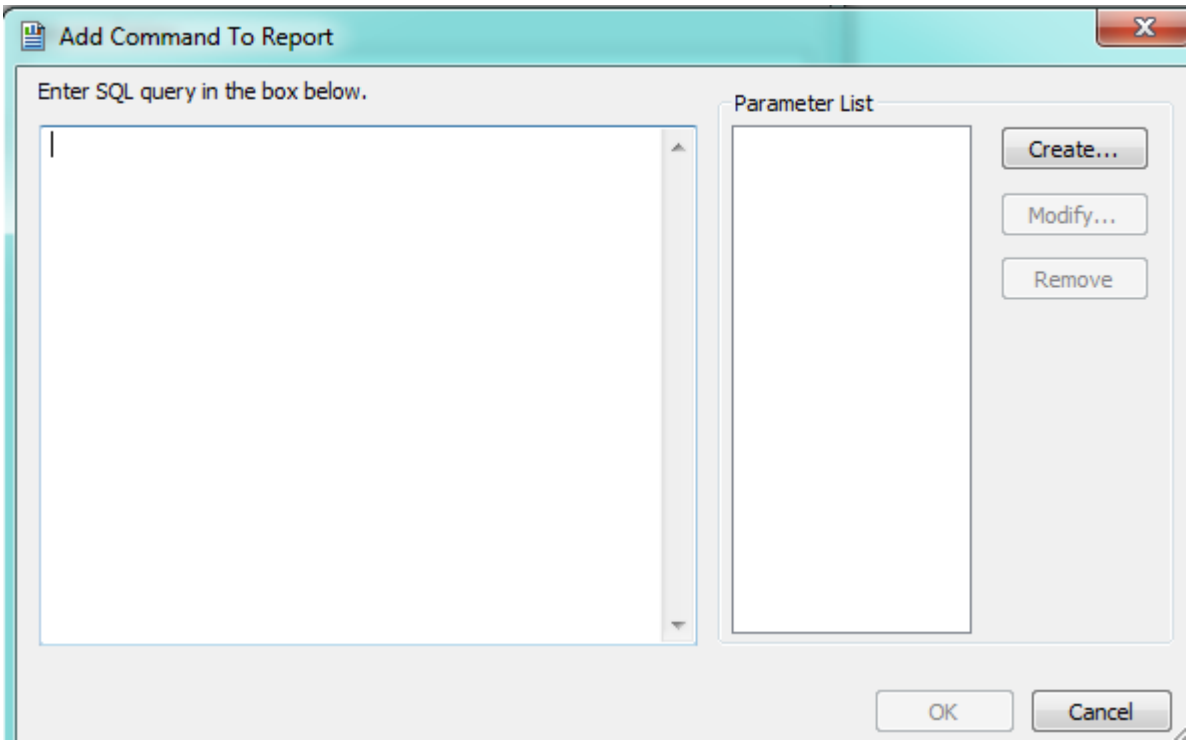
Commands do not recognize Lucity filters when the report is run.

In the previous example for Percent the Total Cost could have been calculated in a Command and been brought in as a field to be used to calculate the percent of the Category cost. This would only have worked with *All Records*.

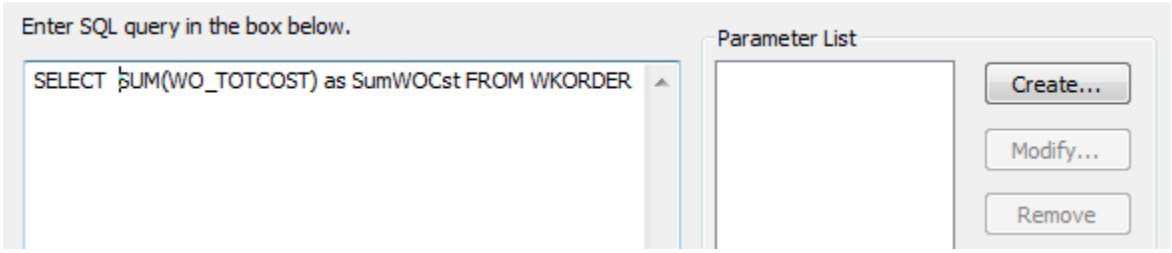
In Database Expert the Add Command would have been selected.



This would open the following window.



The SQL statement would be added to the left box.



The **SumWOCst** field would have been available to use in the percent calculation.

Here is another report which only used a Command.

Task Summary Report - Avg Cost/Unit Sort

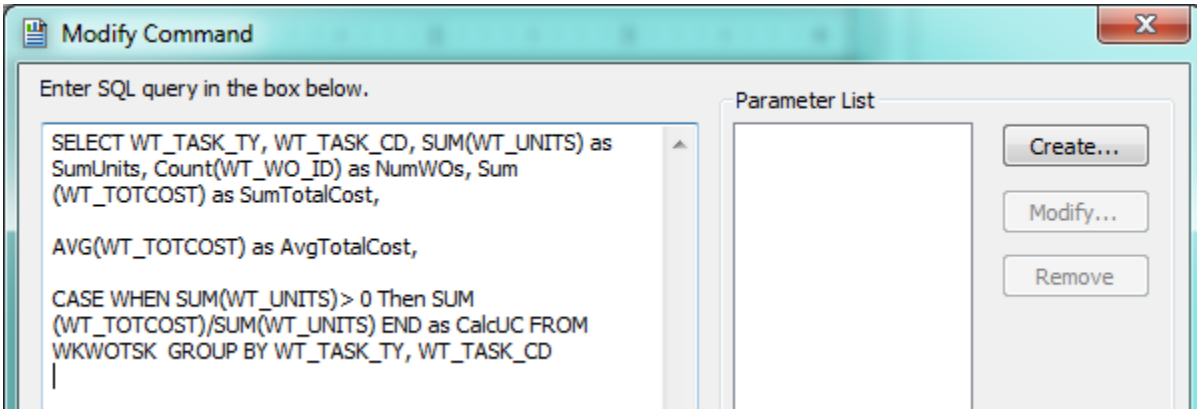
7/17/2014
9:57:54PM

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

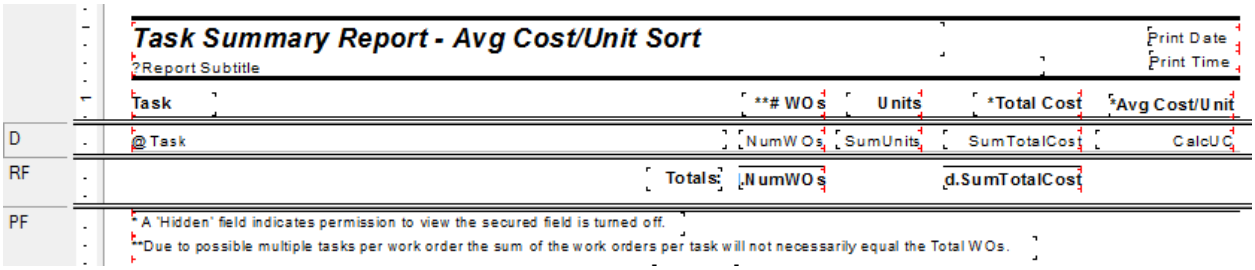
The report needed to group on the Task but the sort was on the **Avg Cost/Unit** formula field. This formula was not available in the *Group Sort Expert* so a Command was used.

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.



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.

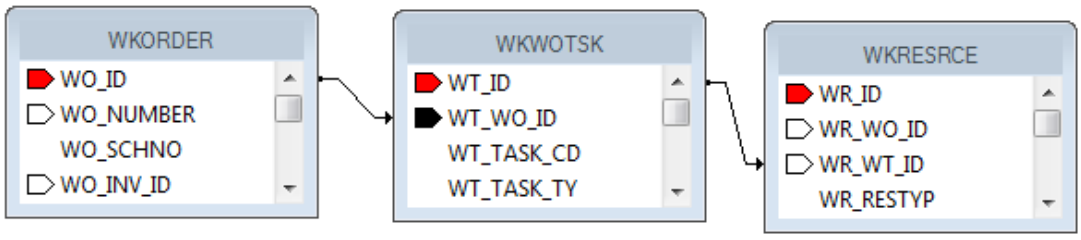
Here is an example of the SQL statement used to create this type of report:

```
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'
```

Table Joins

Usually in Crystal's Database Expert the tables are linked with an outer join. When there is information in the first table then it looks to the table it is joined to with the outer join. This normally shows up with an arrow from the one to the other. (Sometimes in older versions of Crystal the arrow point doesn't show up, so the line looks like an inner join. You have to click on it to figure out the kind of join that is being made.)

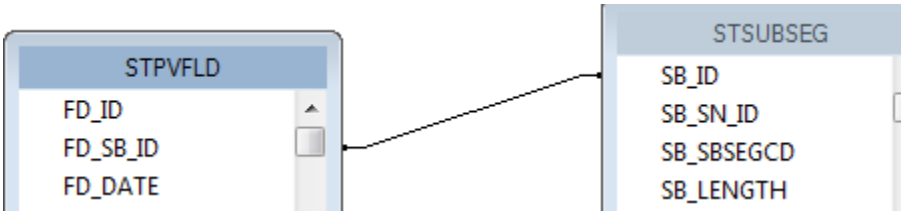
Left Outer Join:



There would need to be a work order before you would pull task information. There would need to be a task before the resource information would show up.

There are some cases within Lucity where an inner join is required. One example is where both Subsegment and Road share the same table (**STPVFLD**) for storing Inspection data. Chances are good that you would not use both Subsegment and Road so this shouldn't be an issue but if you do, then an inner join would be the way to handle the information.

Inner Join:



Now in order to see records both tables would need to have data where **FD_SB_ID** is equal to **SB_ID**.

A similar case is how the Traffic Volume table (**STTVSEG**) connects to Segment and Roads. An inner join is again used.

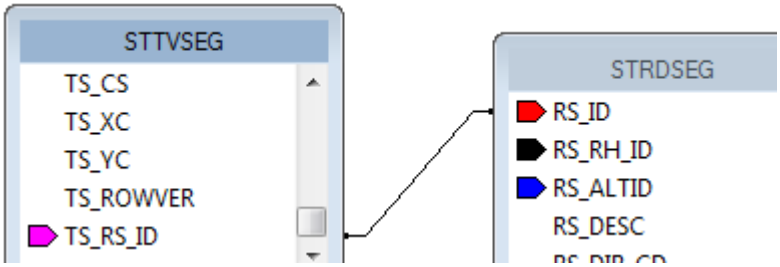
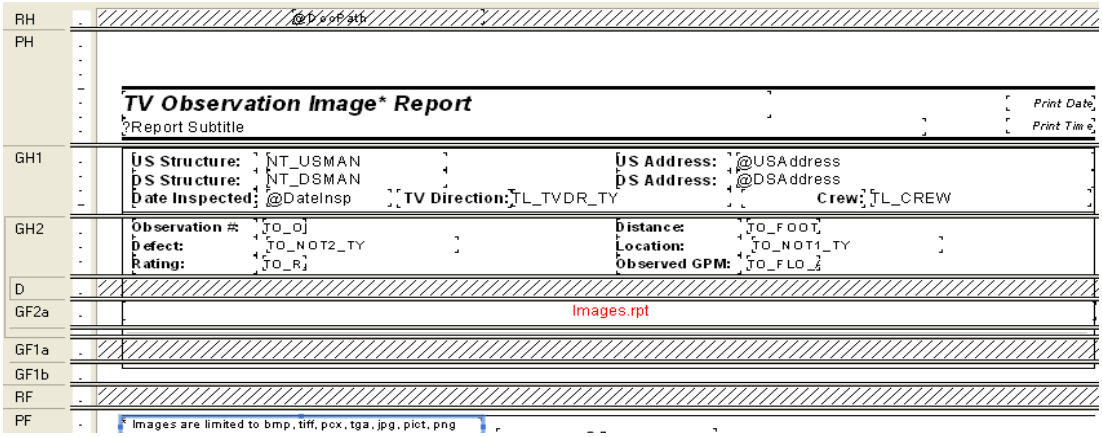
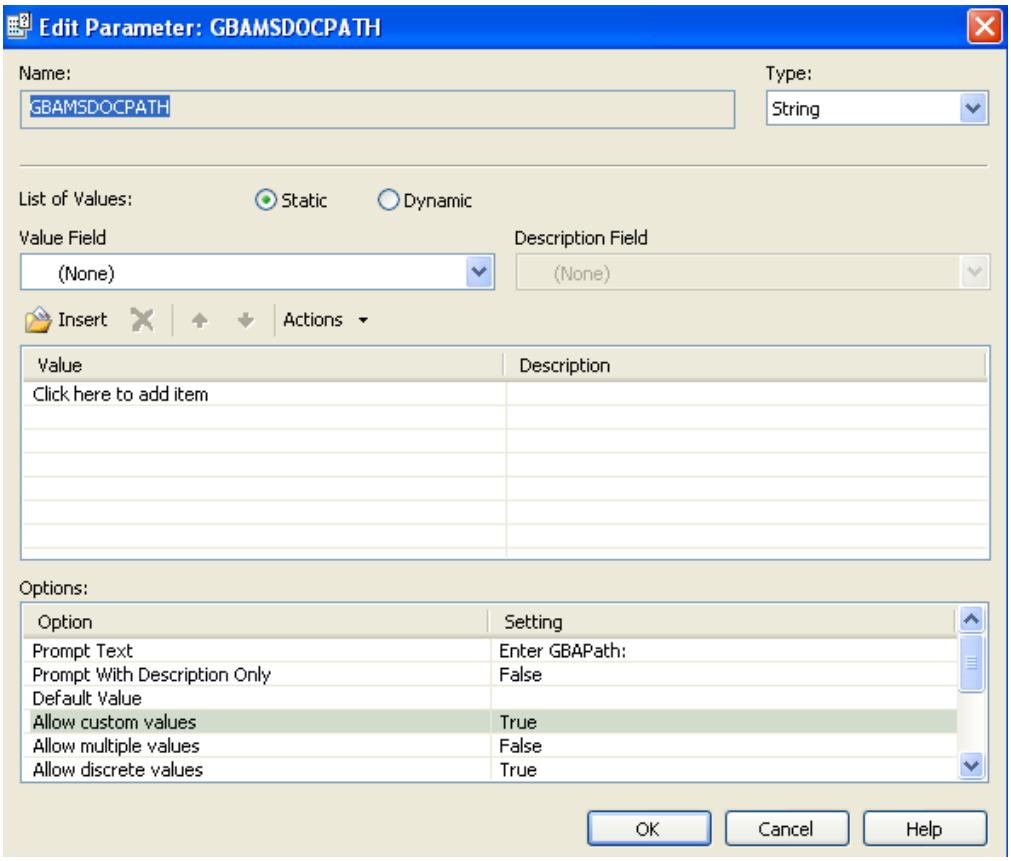


Image Reports

Reports using images can be helpful. There are several imaging reports used throughout the Sewer modules. 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:



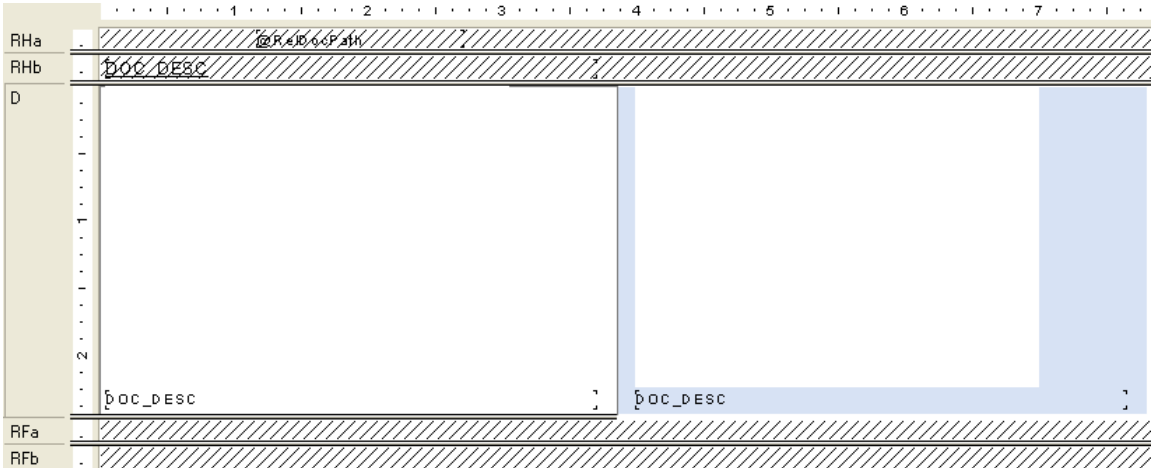
2. Then the **DocPath** formula is created and placed in the *Report Header*:

   shared stringvar GBADocPath:={?GBAMSDOCPATH};

3. The Image subreport is in a *Group Footer*:

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

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



5. The **RelDocPath** formula in the subreport *Header* looks like this:
   shared Stringvar GBADocPath;
   GBADocPath

Notes:_____

_____

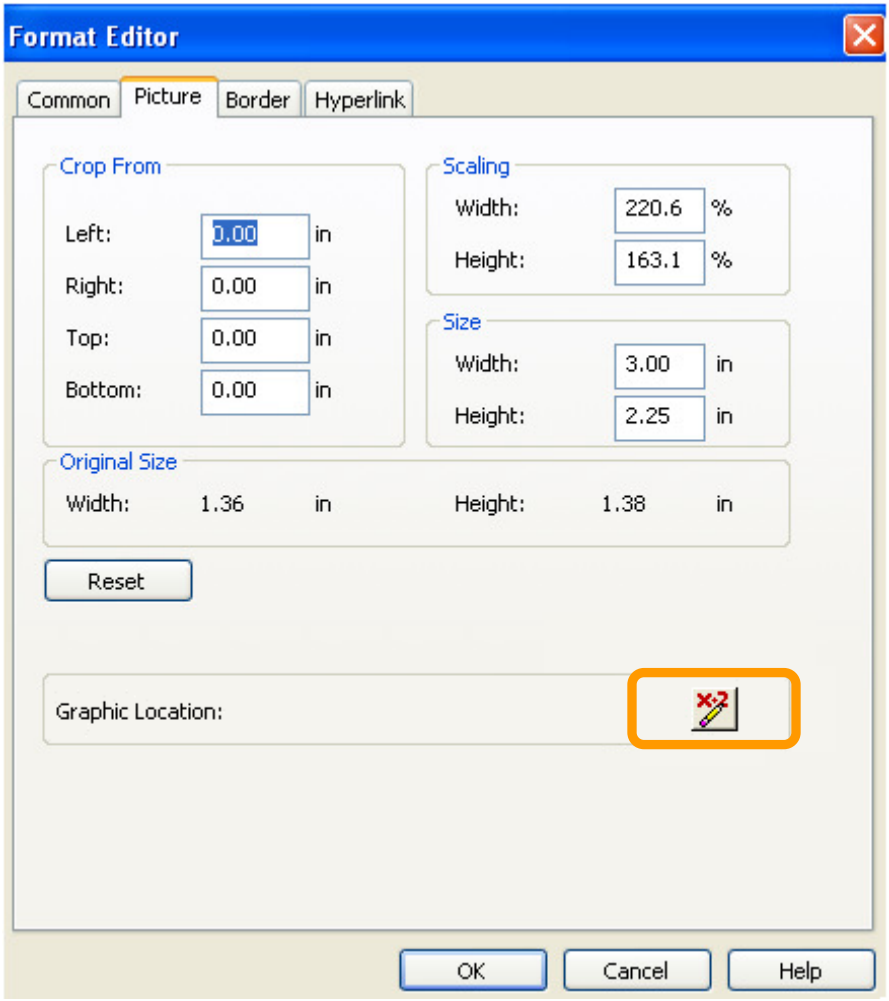
_____

_____

_____

_____

6. 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.



7. 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}
```

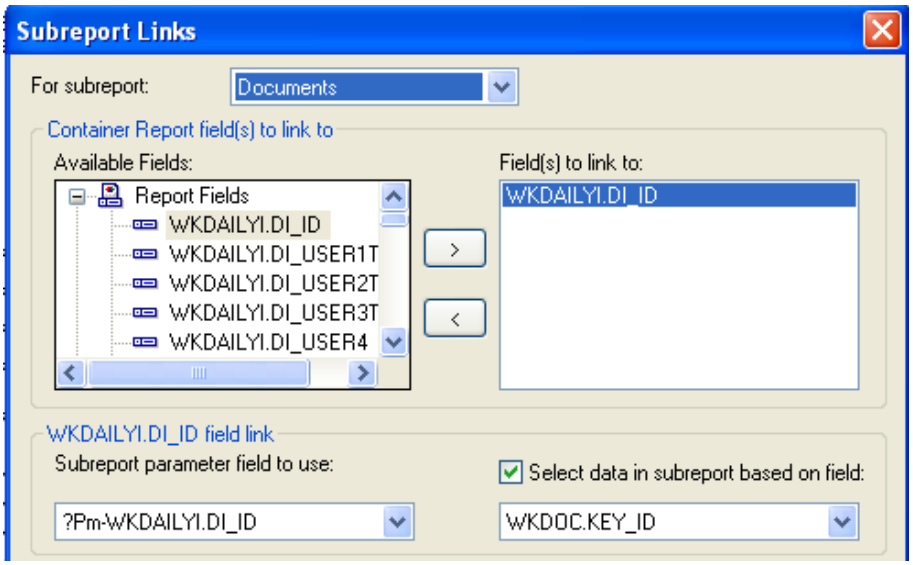
8. To suppress the *Detail* section when there is no image:
   - In *Section Expert* click on the *Details a* section.
   - In the Suppress (No Drill-Down) formula box there is a formula.

```
isnull ({SWDOC.DOC_ID})
```

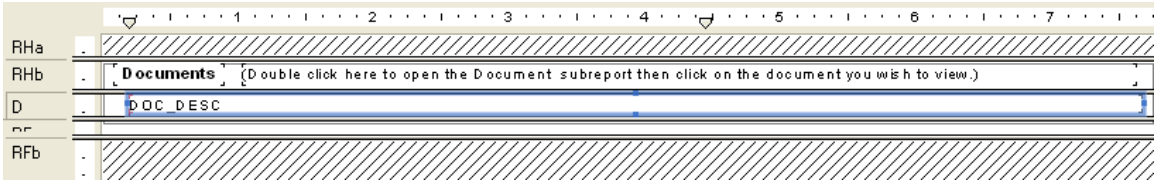
Hyperlinking

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

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



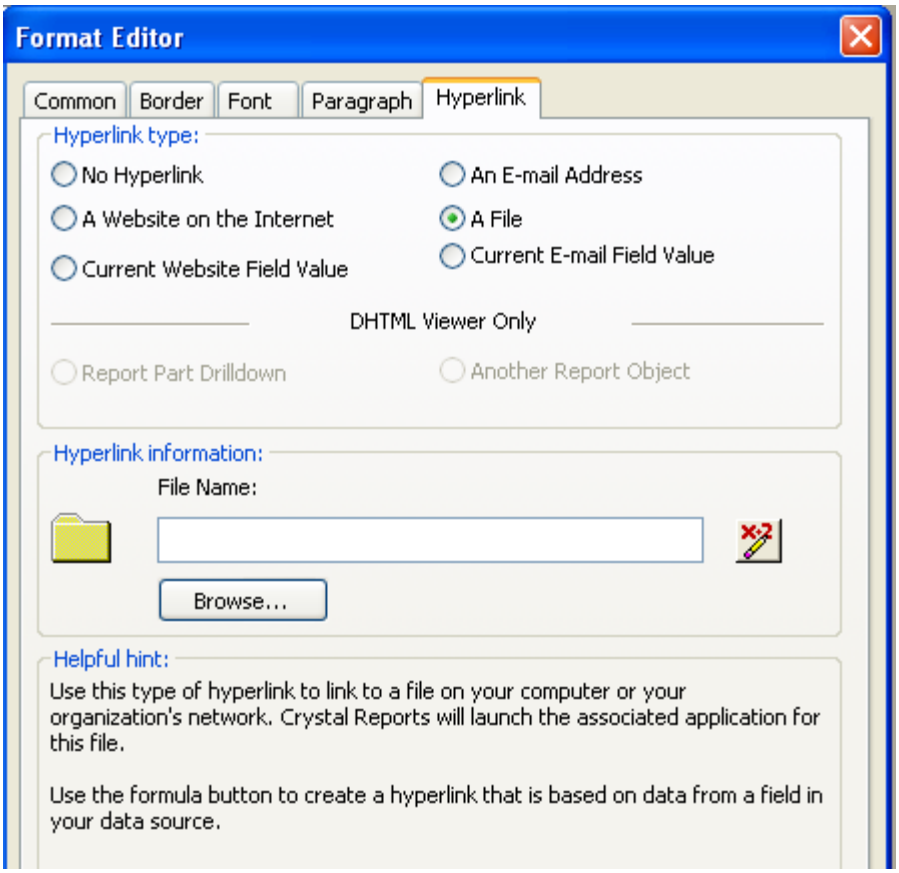
The subreport was set up like this:



- 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.)

Notes:_____

_____

_____

_____

_____

_____

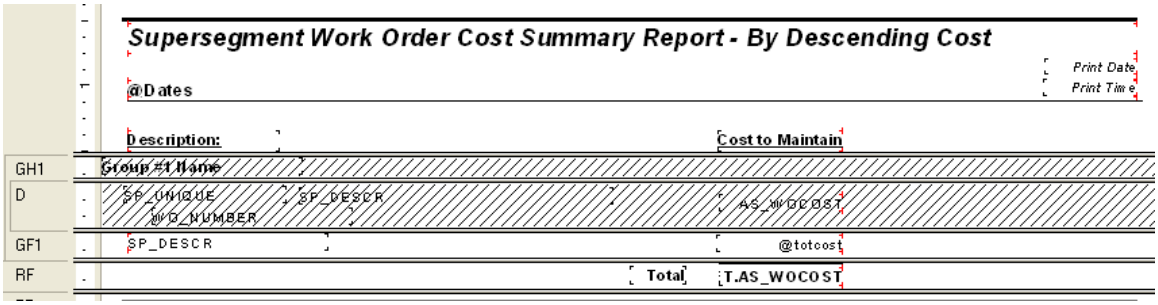
_____

Grouping – More Advanced

Group Sort Expert

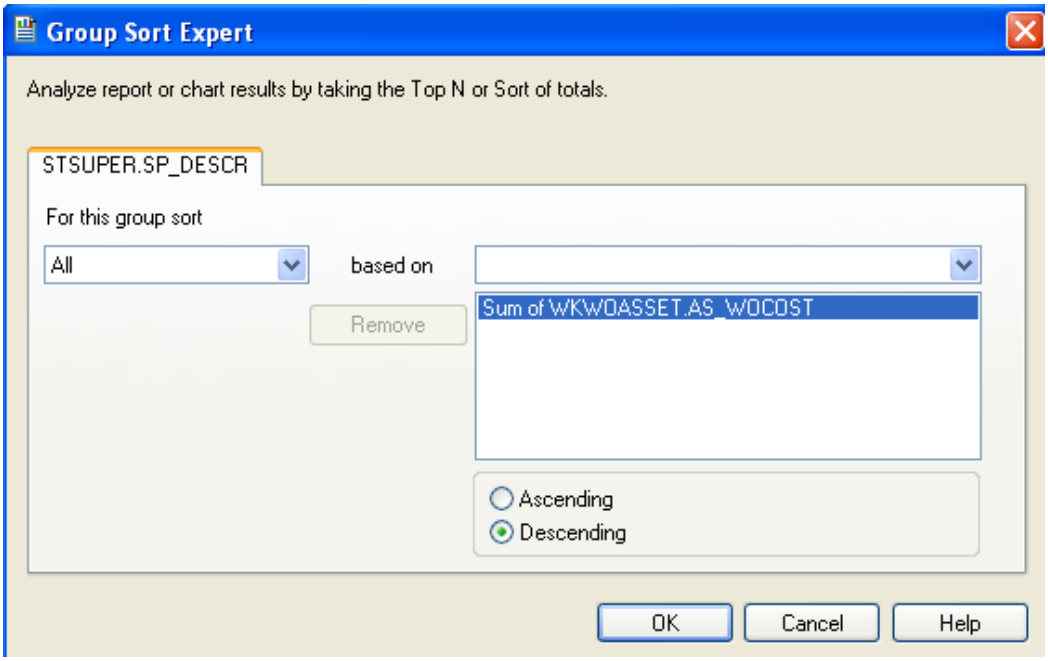
When a report uses grouping, the report automatically sorts on the first group and then within the group any other groups. Each group can be sorted in ascending, descending or a specific order. The *Group Sort Expert* can be used to change the field the report will sort on.

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



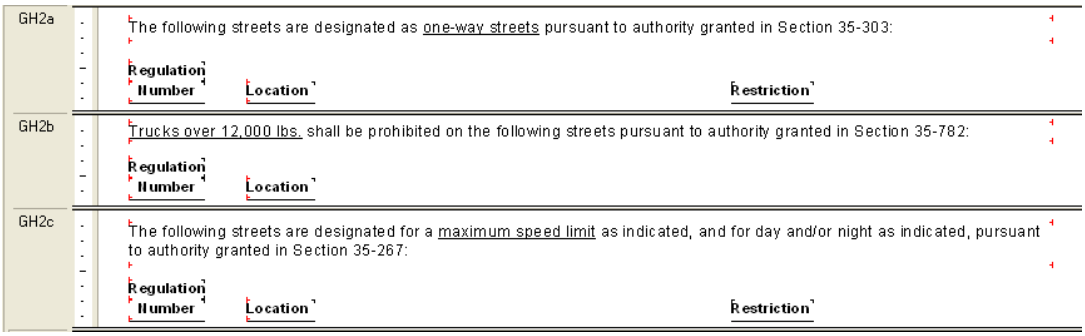
- This report was grouped on the Supersegment description field.

- 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.

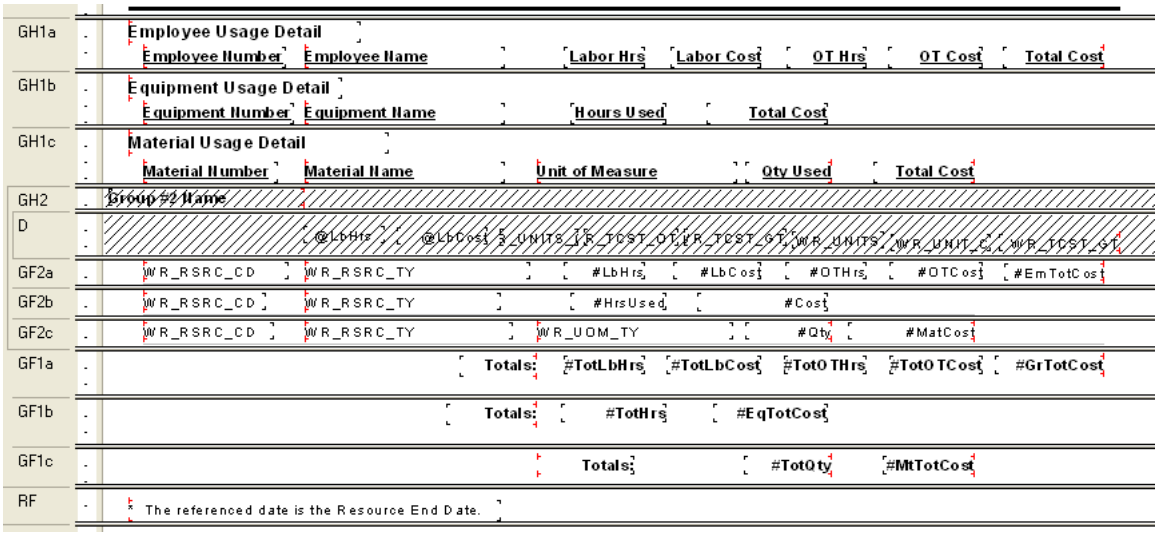
  - *GH2a* suppresses when the following is true:

    {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.

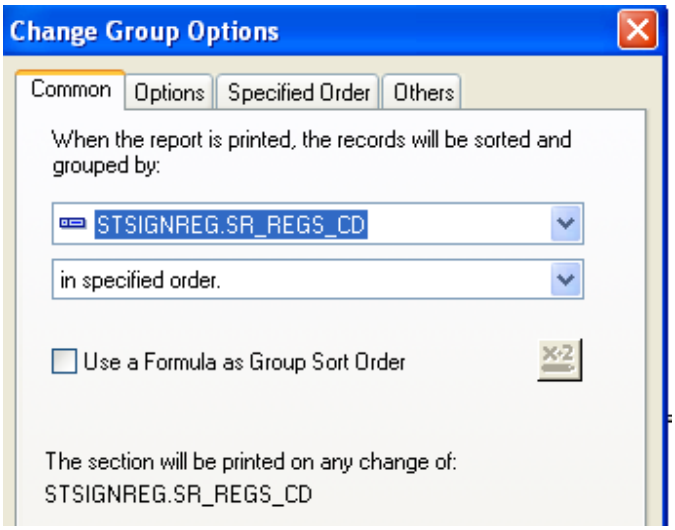


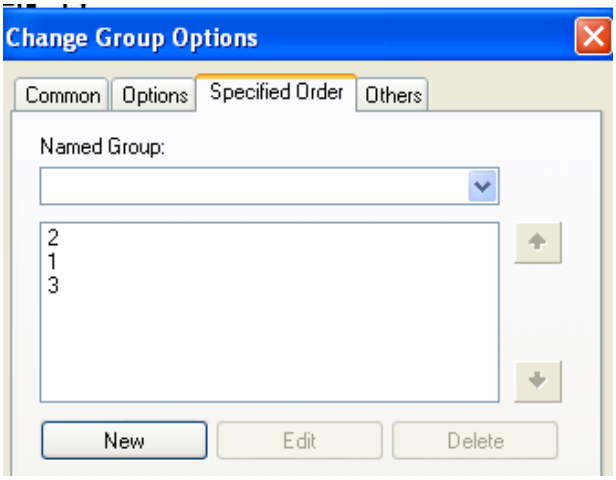
You can use the *Section Expert* to suppress the different *Header* and *Footer* sections. For example the *GH1a*, *GF2a* and *GF1a* all use the following suppression formula:

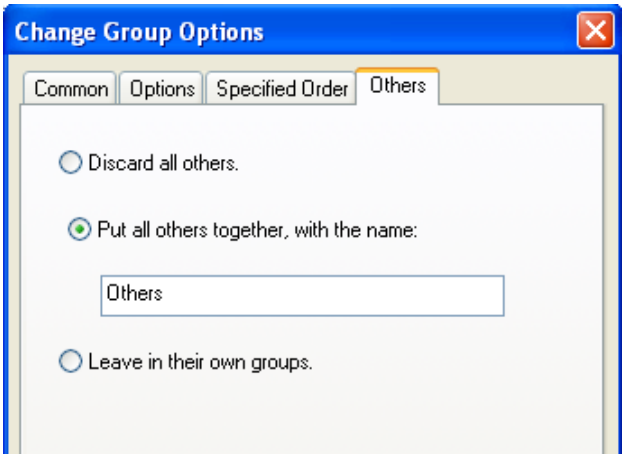
{WKRESRCE.WR_RTYP_CD} in [2,3]

Specific Order of Grouping

In the previous example it was important to have the groups show up in a specific order. This was done using the *Change Group* option. For *Group 2* the following option was used.





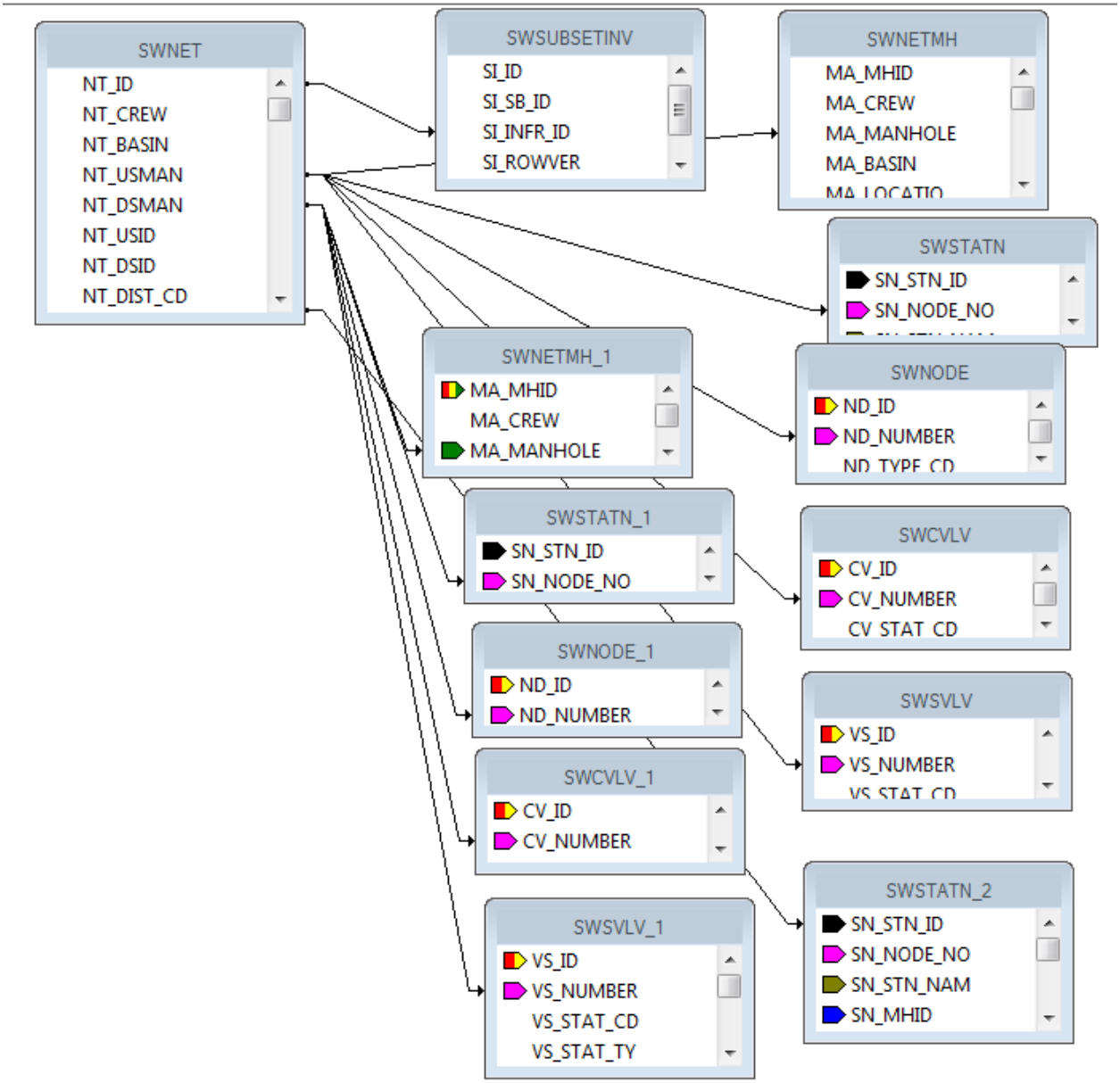


Duplicate Table Use

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)

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,'')

Notes:_____

_____

_____

_____

_____

_____

_____

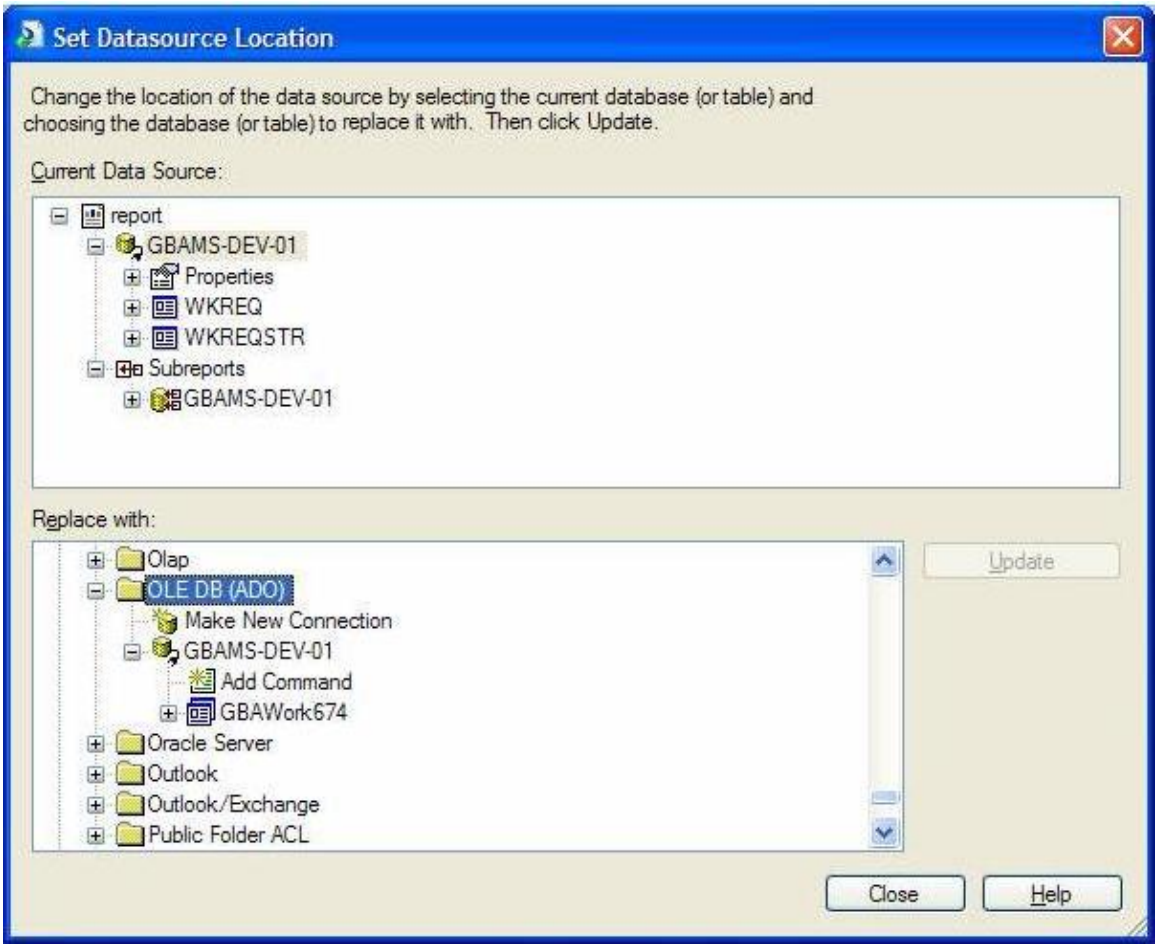
_____

Web Reports

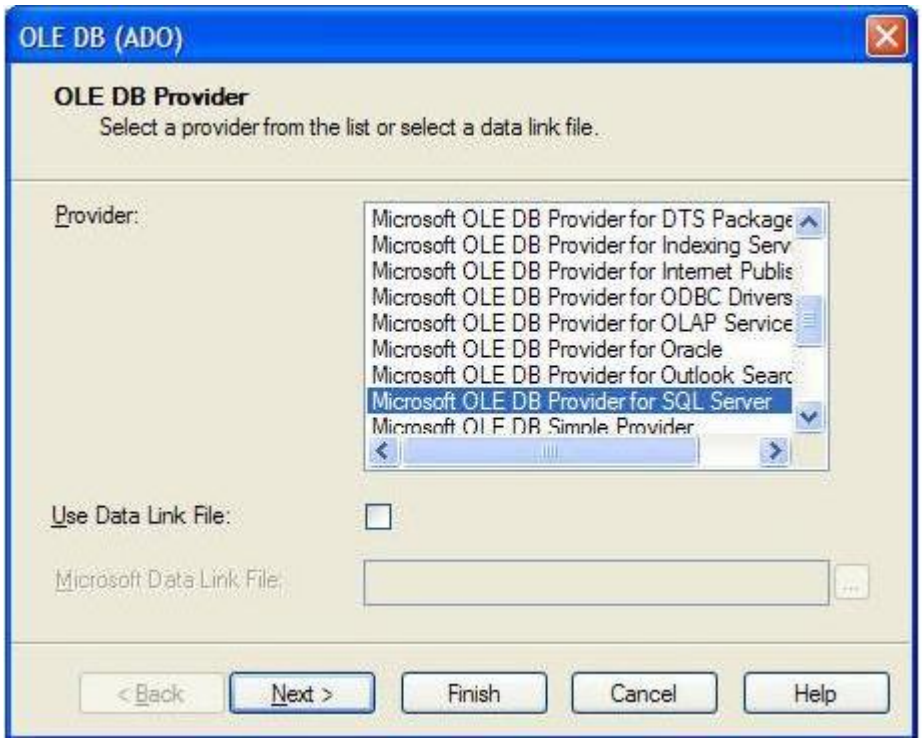
Converting Desktop Reports to use in the Web

Custom Reports that have been created for the desktop version of Lucity will NOT work in the Web application as is. To convert a report from a standard Lucity report to a Web report, complete the following steps:

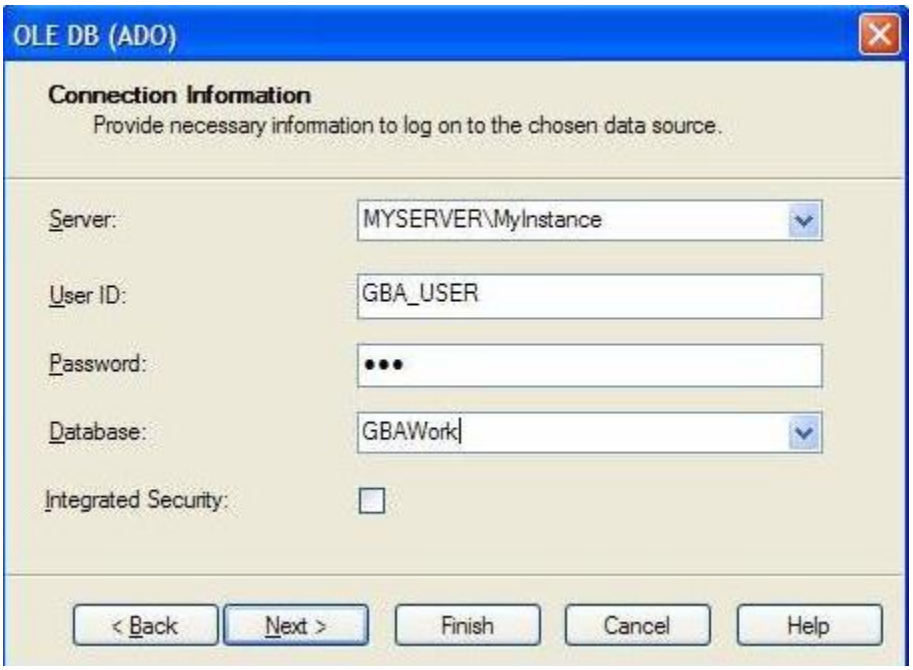
1. Open the report in Crystal Reports.
2. Go to Database>>Set Datasource Location.
    - **SQL Server Clients** select OLE DB (ADO)>>Make New Connection.
    - **Oracle Clients** select Oracle Server.



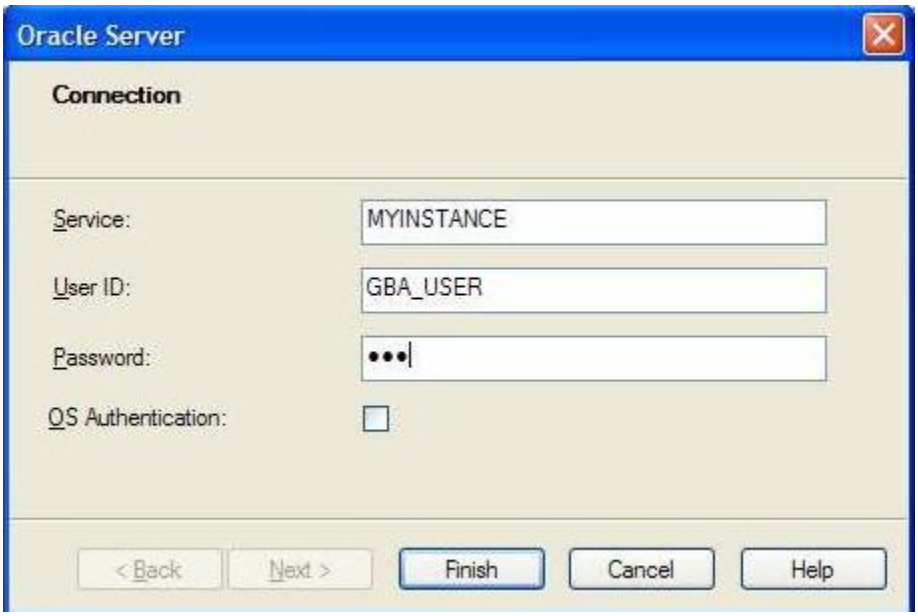
3. If you are using **SQL Server** you will then be prompted to select a provider. Use the Microsoft OLE DB Provider for SQL Server as shown below.



4. Next, enter the Connection information.
    - For **SQL Server**, include the SQL Server Name, User ID, Password, and Database.



- For **Oracle**, include the Service, User ID, and Password.



5. Once you create a connection you will need to individually (manually) map every table from the report one at a time. Don't forget the tables in the Subreports!

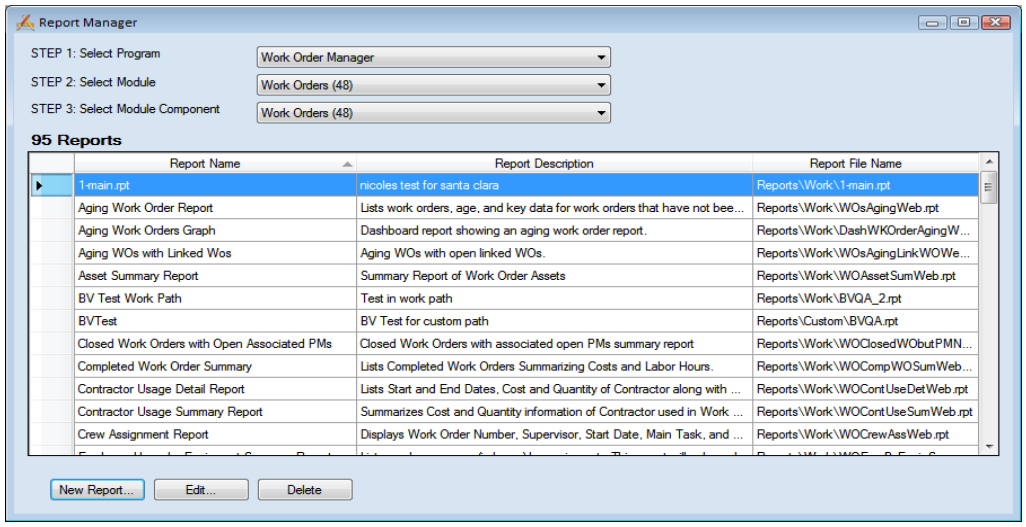
> *Note: Reports must be posted to the web server in the \Reports directory in order to be available online. There you will see sub-directories such as \Equip\ or \Work\. Reports should be placed in the appropriate sub-directories.*

Adding Web Reports

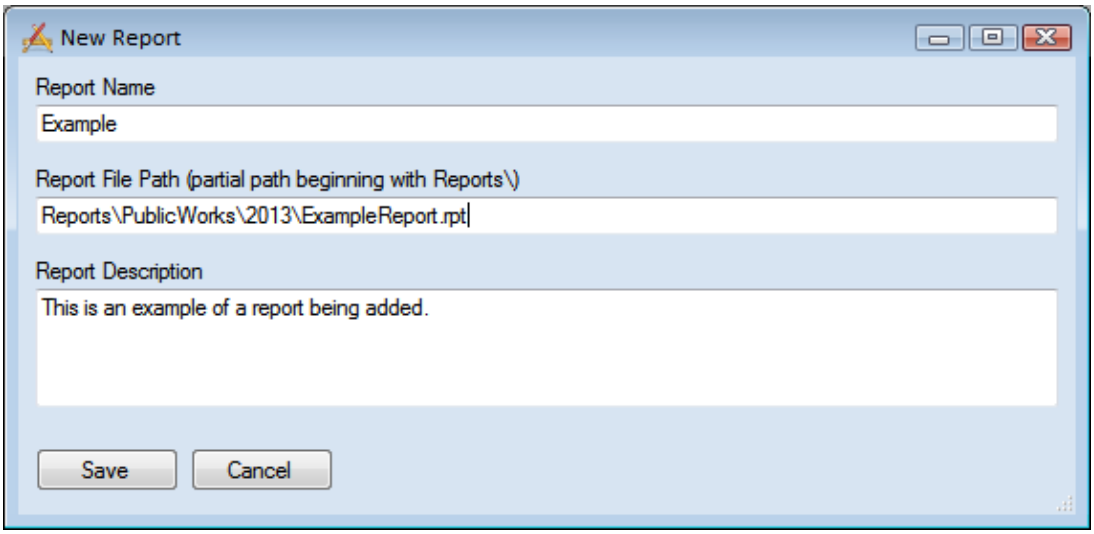
Once a report is converted for Web use it will need to be added to the list of available reports in the module the report is to be run from. There are two different methods for adding reports in the web.

Method 1 - Through the Lucity Administration Tool

1. Launch the Lucity Administration Tool and go to **Reports > Report Manager**.



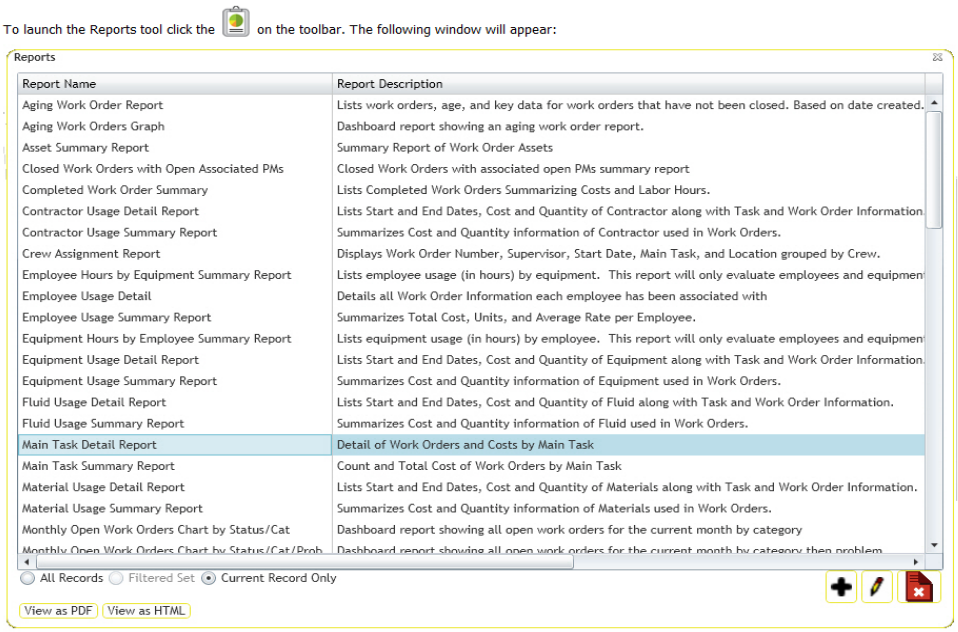
2. Use the drop downs at the top of the tool to navigate to the module that the report is meant for.
3. Click the **New Report** button at the bottom of the Report Manager dialog. The following window will appear:



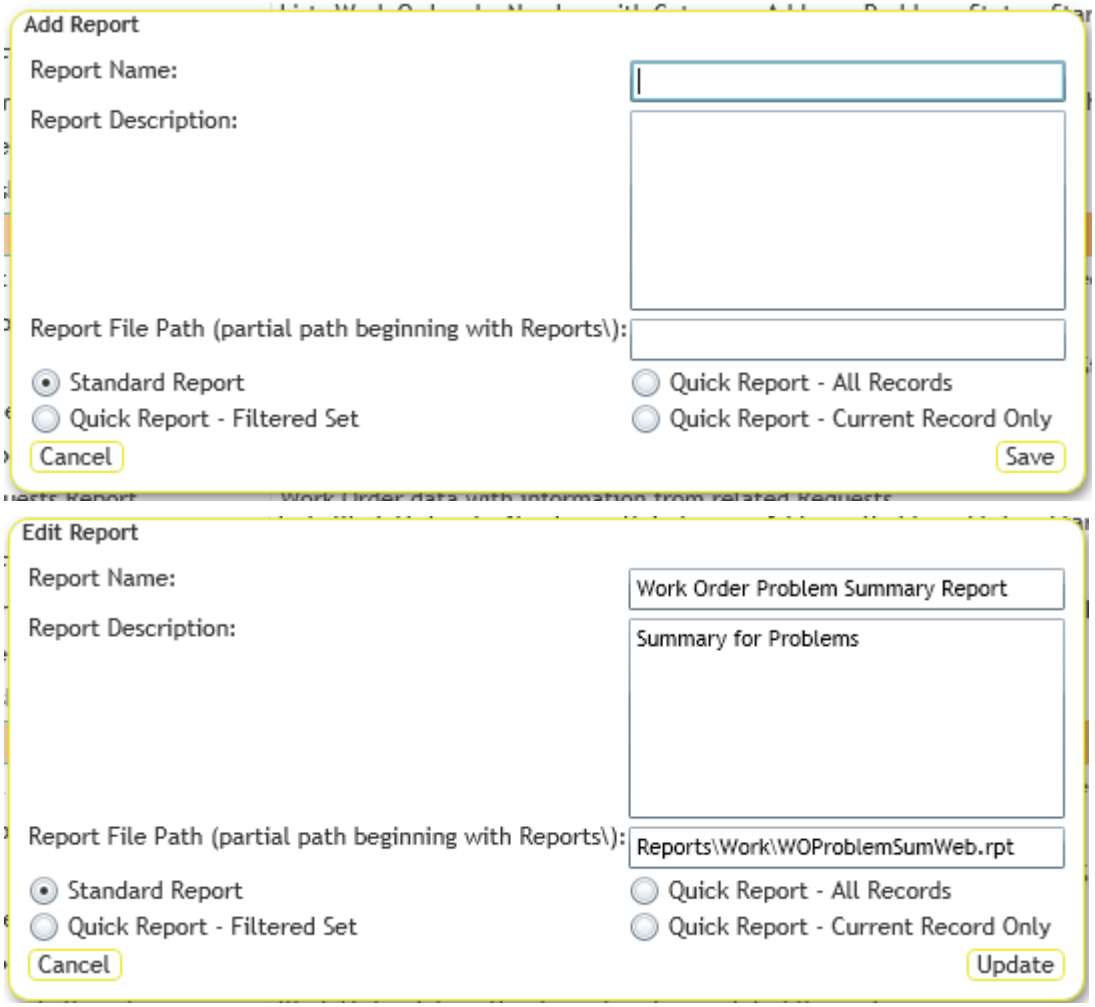
4. Enter the **Report Name**.
5. Enter the **Report File Path**. This should always start with **Reports** and then the rest of the path to the report including the **.rpt** file extension.
6. Enter the **Report Description**.
7. Click **Save**. The new report will be added to the Report Manager.

Method 2 - Through Lucity Web

1. Open the Web module that the report will be run from.
2. Open the Reports Tool.



3. To add a new report, click the **Add button** . The following dialog will appear:



4. Enter a ***Report Name***, ***Report Description***, and the ***Report Filename*** (starting with **Reports**) then the folder name where the report is located followed by a backslash (**Work**) then the name of the actual **.rpt** file. Remember to add the .rpt portion on at the end of the report filename.
5. Choose the type of report this will be
    - Standard Report - The report will be run by clicking the report tool, selecting the report and how to run it, and running it.
    - Quick Report - All Records - Adds the report to the quick report dropdown list next to the report tool. The report will always be run against all records.
    - Quick Report - Filtered Set - Adds the report to the quick report dropdown list next to the report tool. The report will always be run against the current filter set.
    - Quick Report - Current Record Only - Adds the report to the quick report dropdown list next to the report tool. The report will always be run against the current record.
6. Click **Save**.

Web Reports and Parameters

Typically record selection occurs with a filter in Lucity, and then the report is run with this filtered set. Some users wish the report to do the record selection. Reports are able to do this by using Parameters to query the user for the filter that is to be used in selecting records for the report.

The Web reports allow use of parameters but in a limited way.

Parameters set up with a Static list to select from will allow selection of a single value to run the report in either PDF or HTML view.

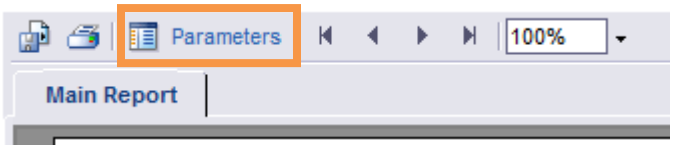
Parameters set up with a Dynamic list to select from will not show a list of values. A single value can be entered manually and must match exactly the field the report is selecting on for either PDF or HTML views.

- A parameter that is set up to allow multiple values will only accept a single value when the report is run.
- The above limitations apply to web reports Viewed as PDF or developed in Crystal XI or earlier. Reports revised or created in Crystal 2008 can be set up to allow the parameter to be queried again in the HTML view so a new parameter value may be selected or multiple values may be chosen (if the parameter field was set up to allow multiple values) and Dynamic parameter lists will be available to select field values.
- The editable parameter option is available in the Parameter setup dialog. The Value Option is "Show on (Viewer) Panel". The Setting options are **Editable**, **Do not show** and **Read Only**.

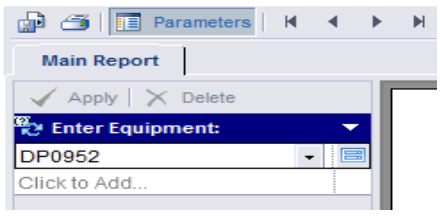
Value Options:

Option	Setting
Show on (Viewer) Panel	Editable
Prompt Text	Enter Equipment:
Prompt With Description Only	False
Optional Prompt	False
Default Value	
Allow custom values	True
Allow multiple values	True
Allow discrete values	True

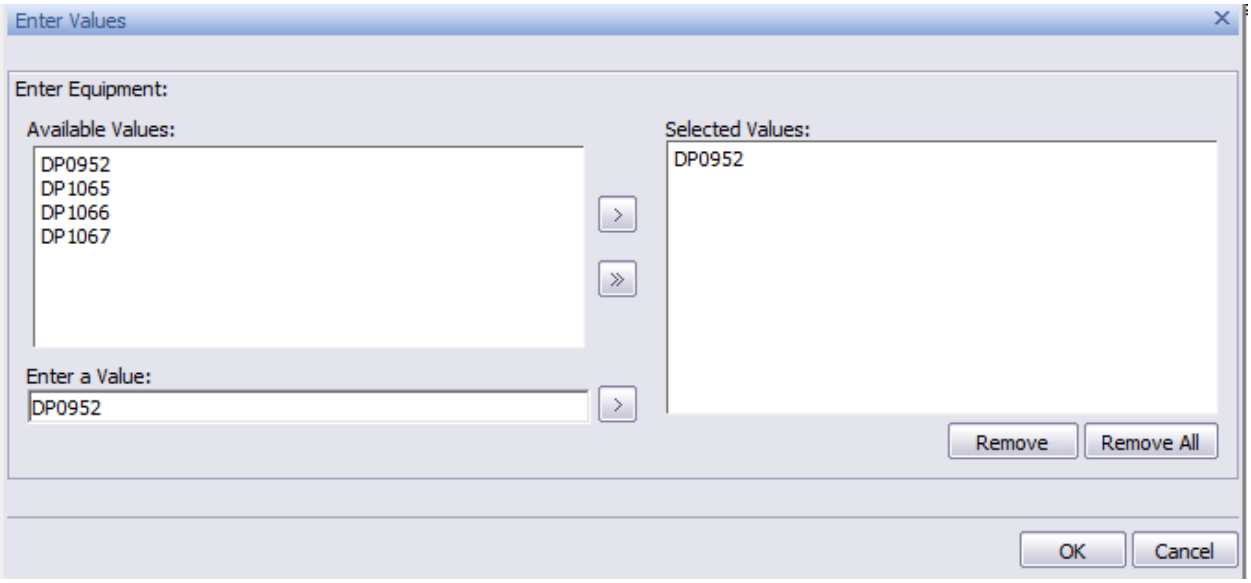
When this setting is **Editable** and the Web report opened with the HTML view, the **Parameter** button will be active for use to modify the Parameter selection. To use this option, complete the following steps:



- Click on **Parameters** to open a panel to the left of the report.



- The actual parameter screen can be opened to either select a new value or multiple values (if the parameter was set up to allow multiples) by clicking on the Parameter value and then clicking the "*Edit parameter value*" button.



- After the new selection values have been set up, click on OK.
- At the top of the Parameter column select "Apply".

> *Note: If parameters are set up in a subreport, they should not have the same name as parameters in the main report.*

Web Date Filters

If a date filter is built in the desktop, the filter uses the "#" sign.

WKORDER.WO_STRT_DT BETWEEN #01/01/2014# AND #12/31/2014#

If this is used in the web and the report is run with 'From Filtered Set', it will error.

Instead use the single quotes in the filter.

WKORDER.WO_STRT_DT BETWEEN '01/01/2012' AND '12/31/2012'

Adding Reports to Dashboard

Standard Reports

To add a standard Dashboard report to your Dashboard, do the following:

1. In *Lucity Admin Tools* click on **Dashboard > Dashboard/Preferences.**
2. If **Lucity** is in the list of users on the left, select it and click **Edit**. If it is not there, click **Add**.
    - If you have to **Add**, then on the next screen select **Lucity**.
3. On the **Dashboard tabs** screen click **Add**.

4. Give the tab a name then click **Add** to add a frame.
5. Give the frame a name then click **Add** to add a plugin.
6. Select **Crystal Report** from the list, give it a name if desired and click **OK**.
7. Select the module you want the report to run in by using the three drop downs.
8. Select the report to run under *Report Name*.
9. To change the filter, click the *Acquire* button. Click **OK**.
10. Click **OK**, click **OK**, click **OK**, click **Save**. Click **Close**.
11. The report should now show in the web dashboard.

Custom Reports

To add a custom report to your dashboard, do the following:

1. The report needs to be converted for the Web.
2. The report name needs to have the word "Dash" at the beginning of the file name.

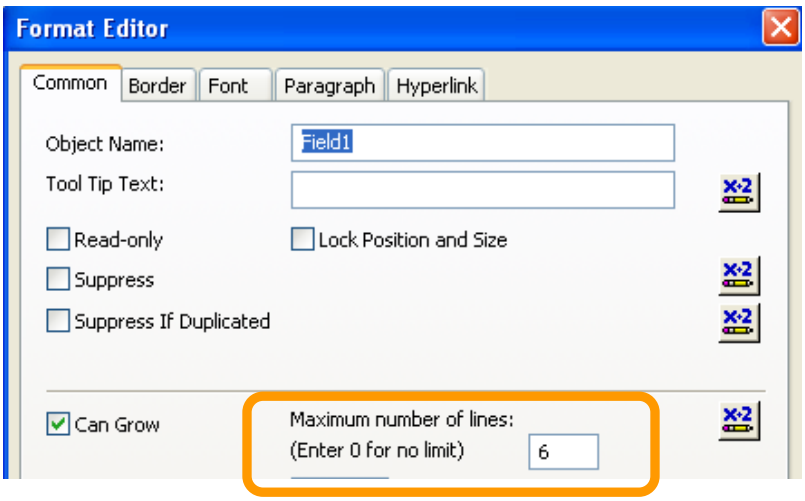
   DashWeeklyWOReportWeb.rpt
3. The report needs to be in the correct Web report folder. (ie Work folder for Work reports)
4. In the Lucity Administration Tools go to Reports > Report Manager and add the Dashboard version of the report as a new report.
5. Then follow the steps listed for adding a Standard Report to the Dashboard.

Odd Things Happen

Field not showing all of the text

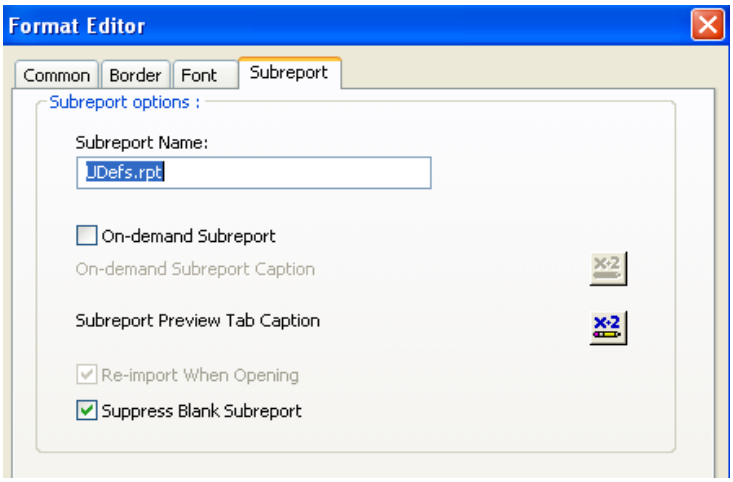
Check the Field Format Editor in the common tab to make sure the *Can Grow* option is checked. If it is checked and the field is still cutting off the data then make sure the maximum lines is set at "0".

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.



Wildcard Use

For Lucity desktop filtering the wildcard symbol is the asterisk "*". If a filter is built in the Lucity web version the wildcard symbol is the percent sign "%". Crystal recognizes the asterisk. The web version converts the symbol before it runs the report. If you typically use the percent sign for a wildcard make sure that when you are working in Crystal that you use the asterisk.

Web Page Number Limitation

When reports are run in the Web with the PDF view the page limitation is ten.

In order to increase the web page limit you will need to add a new line statement to the **appsettings.config** file located on your Web Server in the **C:\inetpub\wwwroot\LucityWeb** folder:

- Open the file with Notepad and paste the following line between the <**appSettings**> and <**/appSettings**>:

  <**add key="MAXPDFPAGES" value="10" /**>

- Adjust the "**10**" to the total number of pages you want the report(s) to account for.

> *Note: the larger the number the longer it will take to generate.*

The file when complete will probably look like this:

<appSettings>

<add key="MAXPDFPAGES" value="50" />

</appSettings>

This may require an IIS reset to take the changes.