



TRAINING GUIDE

Advanced Crystal 2



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)	3
Variables	6
Why Use Variables?	6
Declaring a Variable	7
Assigning a Value to a Variable	8
Declare and Assign	8
Evaluation Time	8
Using Variables in a Report	8
Unlinked Subreports	10
Date Variables	12
Using Variables for Subtraction	12
Using Variables to Count for Section Suppression	14
Using Variables and Documentation	15
Comments to Non Comment Subreports	17
Option 1 - Variables	17
Option 2 - Grouping & Conditional Suppression	19
Percentages	20
Percentage with Insert Summary	20
Percentage with a Formula	21
Percentage with Variables	22
Use of Commands	24
Image Reports	27
Hyperlinking	30
Grouping - More Advanced	32
Group Sort Expert	32
Section Expert and Grouping	33
Specific Order of Grouping	34
Duplicate Table Use	36

Conversions	37
Web Reports	38
Converting Desktop Reports to use in the Web	38
Adding Custom Web Reports (Prior to Version 2015)	41
Method 1 - Through the Lucity Administration Tool	42
Method 2 - Through Lucity Web	43
Custom Web Reports Version 2015 (sp1) and Beyond	44
Understanding the Process	45
How Lucity Manages Custom Reports	45
Best Practice for Making the Change	47
Setup	47
How To Add New Custom Reports	48
Editing Reports	50
Updating Custom Reports	50
Delete Report	51
Report Security	52
Web Report Grouping	53
Web Reports and Parameters	54
Dynamic Parameters	56
Web Filters	56
Date	56
Wildcard Use	56
Adding Reports to Dashboard (prior to V2015r2)	57
Standard Reports	57
Odd Things Happen	57
Field not showing all of the text	57
Report Takes a Long Time to Open	58
Web (PDF) Page Number Limitation	58
Prior to V2015r2	58
V2015r2	59

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
<small>Double click the category code to view the individual work orders.</small>		
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
2008-01862	Complete	7/25/2006	Equipment	Miscellaneous	\$156.59
2008-01882	Complete	8/7/2008	Equipment	Dist Maintenance/Equipment	\$155.47
2008-01884	Complete	8/7/2008	Equipment	Miscellaneous	\$208.35
2008-01885	Complete	8/7/2008	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. Open the new report and 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.

PHa	Work Orders by Category Report					Print Date
	Report Subtitle					Print Time
	Category					
PHb	WO #	Status	Status Date	Main Task	*Total Cost	
GH1	Group #1 Name					
GH2	Group #2 Name					
D						
GF2a	WO_NUMBER	WO_STAT_TY	@StatusDate	WO_ACTN_TY	@CostTot@WOCost	
GF1						
RF	@GrTot					

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.

Work Orders by Category Report							Print Date
Report Subtitle							Print Time
Category						Count	
GH1	Group #1 Name						
	WO #	Status	Status Date	Main Task			Total Cost
GH2	Group #2 Name						
D	[Hatched Section]						
GF2a	WO_NUMBER	WO_STAT_TY	@Status Date	WO_ACTN_TY			@CostTot @ WOCost
GF1							
RF							

12. Create a count formula using the *Summary* button  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.

Insert Summary X

Choose the field to summarize:

Calculate this summary:

Summary location
 Insert Group...

Options
 Show as a percentage of

 Summarize across hierarchy

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.

PH	Work Orders by Category Report					Print Date
	?Report Subtitle					Print Time
	Category				Count	
GH1	Group #1 Name	Status	Status Date	Main Task	Total Cost	
GH2	Group #2 Name					
D	WO #	WO_STAT_TY	Status Date	WO_ACTN_TY	Cost	WOCost
GF1	@Category				WO_NUMBER	
RF					@GrTot	

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	
80000 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	\$158.59
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	

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:

Work Order Category Summary Report				Print Date
?Report Subtitle				Print Time
Category			Count of WO's	Total Cost
GH1				
D				
GF1	{WO_CAT_CD}	{WO_CAT_TY}	{DER.WO_ID}	{R.WO_TOTCOST}
RF	{Grand Totals:}		{RDER.WO_ID}	{R.WO_TOTCOST}

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	\$2,598.80
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

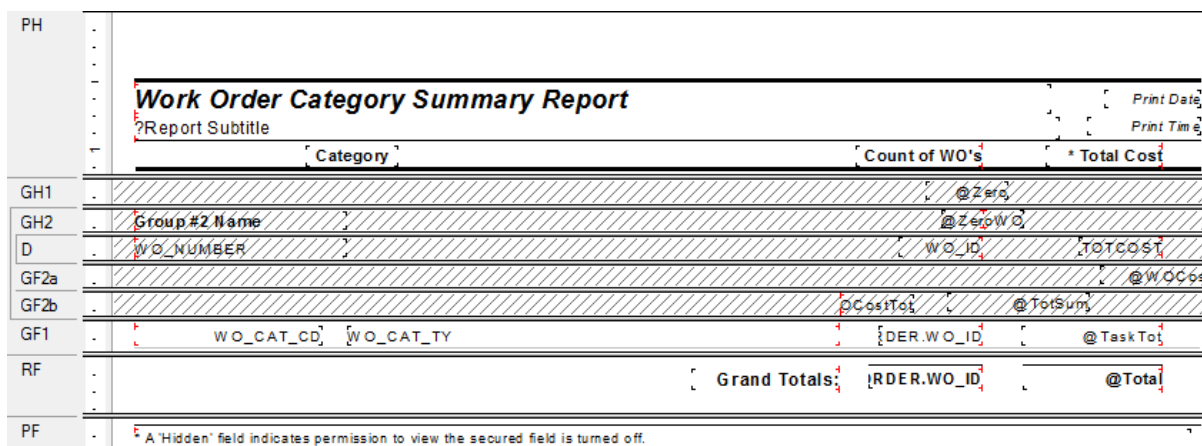
Evaluation Times are statements that tell when the formula is to be evaluated. If nothing is stated, Crystal Reports guesses what is appropriate for the data being used in the formula.

- **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

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



Category	Count of WO's	* Total Cost
Group #2 Name	@ZeroWO	@ZeroWO
WO_NUMBER	WO_ID	TOTCOST
	@CostTot	@TotSum
WO_CAT_CD	WO_CAT_TY	{DER.WO_ID} @TaskTot
Grand Totals:	{RDER.WO_ID}	@Total

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

- 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}
```

- 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
```

- 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
```

- 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;
```

- 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;
```

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

```
WhilePrintingRecords;  
Shared numberVar WOCostTot ;  
WOCostTot
```

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

```
WhilePrintingRecords;  
Shared numberVar GrWOCost ;  
GrWOCost
```

- 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. 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 *Lucy* 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.

Lucy 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.



RHa	-	
RHb	-	
D	-	@StoreUserBtns
RFa	-	
RFb	-	

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.

Flow Meter Location Usage Report		Print Date
Report Subtitle		Print Time
@Dates		
GH1	Meter Loc Number: Group #1 Name	@ZeroLoc Address: @Address/Apt
GH2a	Meter Device Number: MD_NUMBER	@ZeroDev Group #2
GH2b	@ZeroPrev Date Reading 1 Amount 1	
Da	@Date RM_READ1	@Diff1
Db	@PrevVal @CalcTot1	
GF2	Device Total: @TotMet1	
GF1	Location Total: @TotLoc1	
RF		

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 Numbevar 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:


```

      (PrevVal)
      Shared Numbevar prevval1;
      prevval1:={WTMTMD.RM_READ1};

      (CalcTots)
      shared Numbevar MetTot1;
      shared Numbevar LocTot1;
      MetTot1:=MetTot1 + {@Diff1};
      LocTot1:=LocTot1 + {@Diff1};
      
```
- In *GF2* there is a formula for Device total usage (**TotMet1**):


```

      shared Numbevar MetTot1;
      MetTot1
      
```
- In *GF1* there is a formula for the Location total usage (**TotLoc1**):

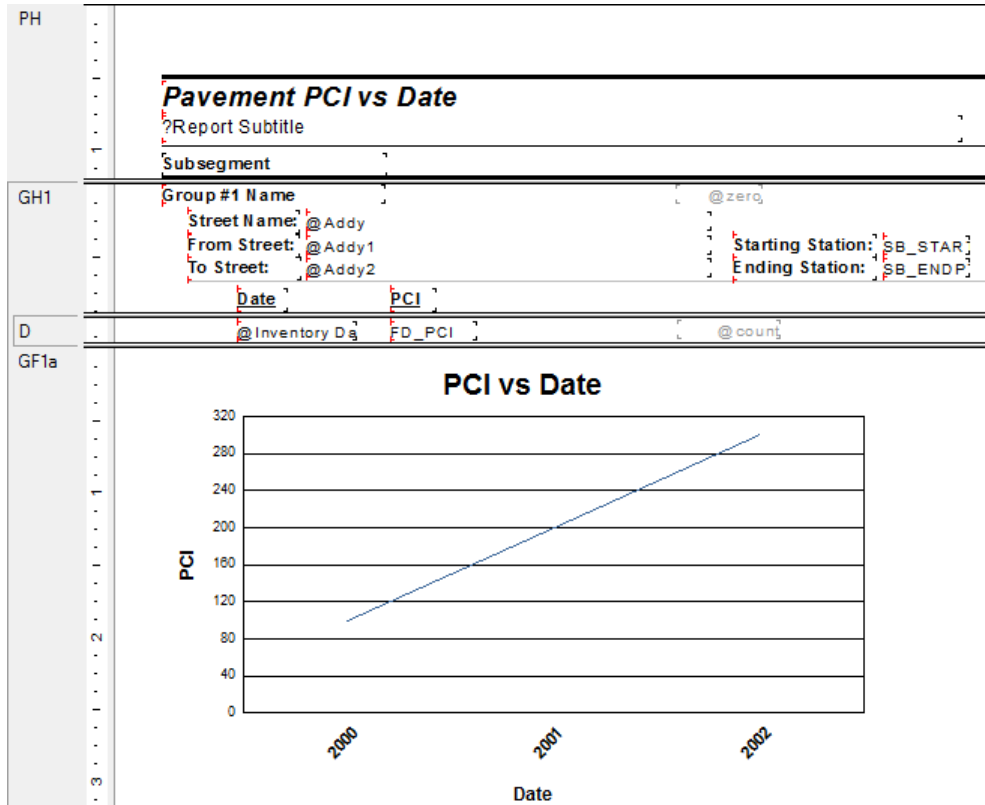

```

      shared Numbevar LocTot1;
      LocTot1
      
```

Note: _____

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.

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} & " ,";
```


Response Time Summary Report											
Report Subtitle											
		Office Hours				After Hours					
		Total Responses	Responses with in 1 Hr.	Responses between 1 & 2 Hrs.	Responses greater than 2 hrs.	Total Responses	Responses with in 1 Hr.	Responses between 1 & 2 Hrs.	Responses greater than 2 hrs.	# Svc not restored within 12 hrs.	
Category	Problem										
GH1	Group #1 Name The user wanted the report grouped by category and then by problem (for request)										
GH2	Group #2 Name @Rec@resq										
GH3	Group #3 Name @Zero										
Da	Request										
Db	event										
Dc	@DateTest @MinR @MinE @EventDt @MinR @MinE										
Dd	@EventDt @DateTest @Afterhrs @srvnote @Req @norequest @MinD @After @Min @Diff @negw @res										
GF3											
GF2a	WO_CAT_TY	WO_PROB_TY	hourR	w60m	zmin	zmin1	@Req	{Min	{wmin	{wmin1	{serv
GF2b	WO's with negative times (counts included): @shownegw@time										
GF2c	WO's with no request (counts not included): @shownoreques										
GF2d	int_totals										
RF		{total}	{Htotal}	{Htotal}	{Htotal}	{total}	{Htotal}	{Htotal}	{Htotal}	{restota}	

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.

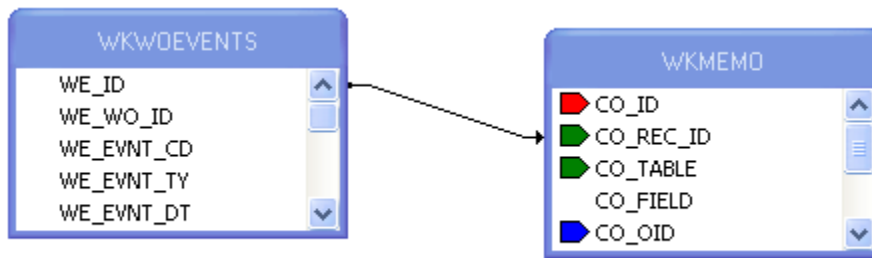
There are a variety of ways to accomplish this task. Not all work for the various scenarios.

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:

Event	Date	Time	Note
Group #1 Name			
WE_EVNT_TY	@Date	@Time	WE_NOTE
Comments	@CommentTxt		

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}
```

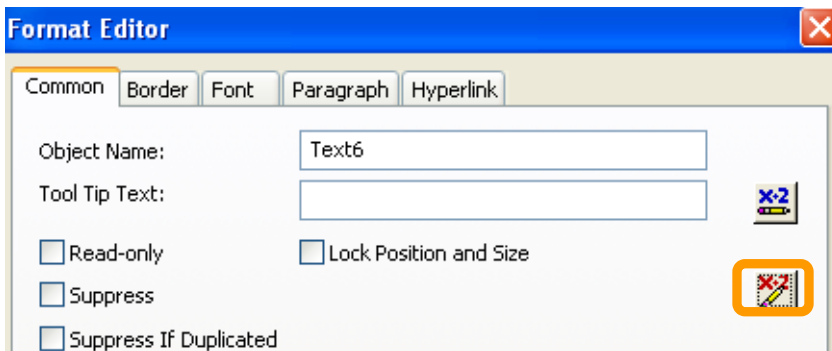
4. Create a reset formula (**Zero**) for this variable and place it in the *Group Header* section. The zero in this case is a space “ ” instead of a zero (0) because it is a string variable.

```
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.

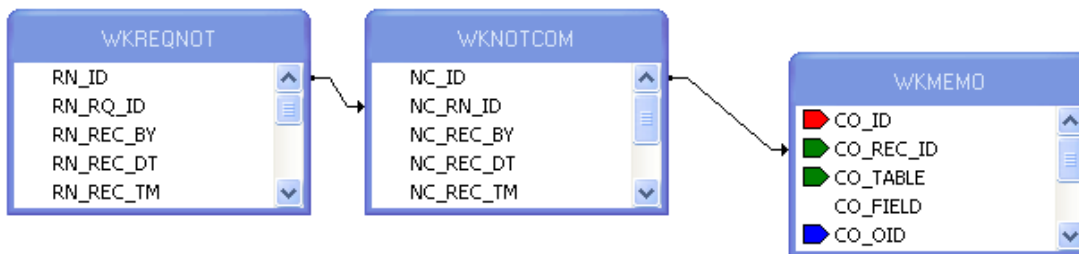
The comment was added to the *Notifications* subreport within the **Request Detail Report (ReqDetail.rpt)**.

Notifications					
Initiated Date /	Initiated By	Agency	Contact	Purpose Text	
10/20/2009	GBA	DEF	Ted		
10/20/2009	GBA	ABC	Babs	Follow Up	

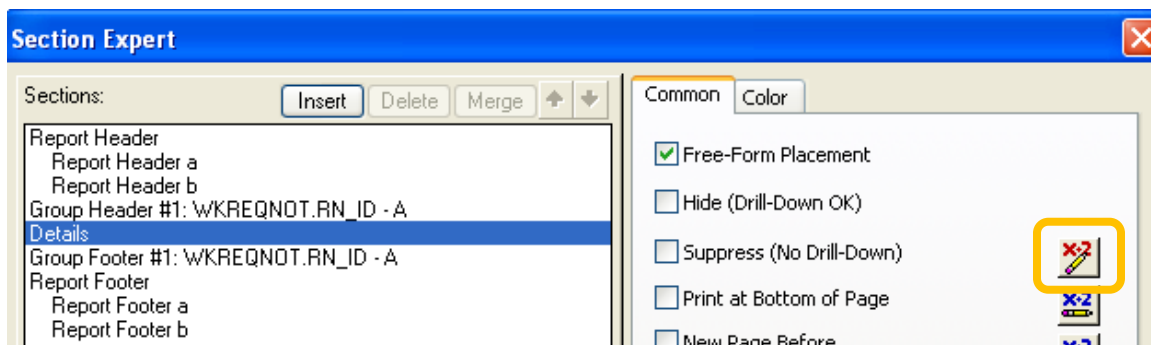
Notification Comments	
Recorded Date /	Recorded By
10/20/2009	GBA

RHa	/
RHb	Notifications
	Initiated By Date Time Agency Contact Purpose
GH1	RN_REC_BY @Date @Time RN_AGENCY RN_CONTACT RN_PURP_TY
D	@Com Date Time NC_REC_BY CO_TEXT
RFa	/
RFb	/

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.
 - 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.
 - o Right align at 7”.
4. Calculate the value for the Category Cost as a percent of the Total Cost.
 - o Click on Insert Summary. Set up as follows and click *OK*.

The screenshot shows the 'Insert Summary' dialog box. The 'Choose the field to summarize:' dropdown is set to 'WKORDER.WO_TOTCOST'. The 'Calculate this summary:' dropdown is set to 'Sum'. The 'Summary location:' dropdown is set to 'Group #1: WKORDER.WO_CAT_CD - A'. The 'Options:' section has the 'Show as a percentage of' checkbox checked, and the 'Grand Total: Sum of WO_TOTCOST' dropdown is visible below it. A red oval highlights the 'Show as a percentage of' checkbox and its label.

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

Work Order Category Summary Report				
Report Subtitle		Count of WO's	* Total Cost	Percent
GH1				
GH2	Group #2 Name	@ZeroWO	@Zero	
D	WO_NUMBER	WO_ID	TOTCOST	
GF2a			@CostTot	@WOCost
GF2b			@CostTot	@TotSum
GF1	WO_CAT_CD, WO_CAT_TY	{DER.WO_ID}	@TaskTot	TOTCOST
RF		Grand Totals: {RDER.WO_ID}	@Total	

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	\$2,598.80	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 %	38.71 %
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.
 - a. Right click on *Running Total Fields*.
 - b. Click *New*

- c. Give it a name (**TotalCost**).
- d. Set up as follows:

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

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

- i. Click *Save and close*
- j. Drag the formula into the *Report Footer*.
- k. 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.

- a. Right click on the **CatPercV** formula and select *Edit*.
- b. 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.

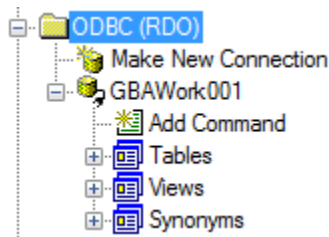
Use of Commands

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.

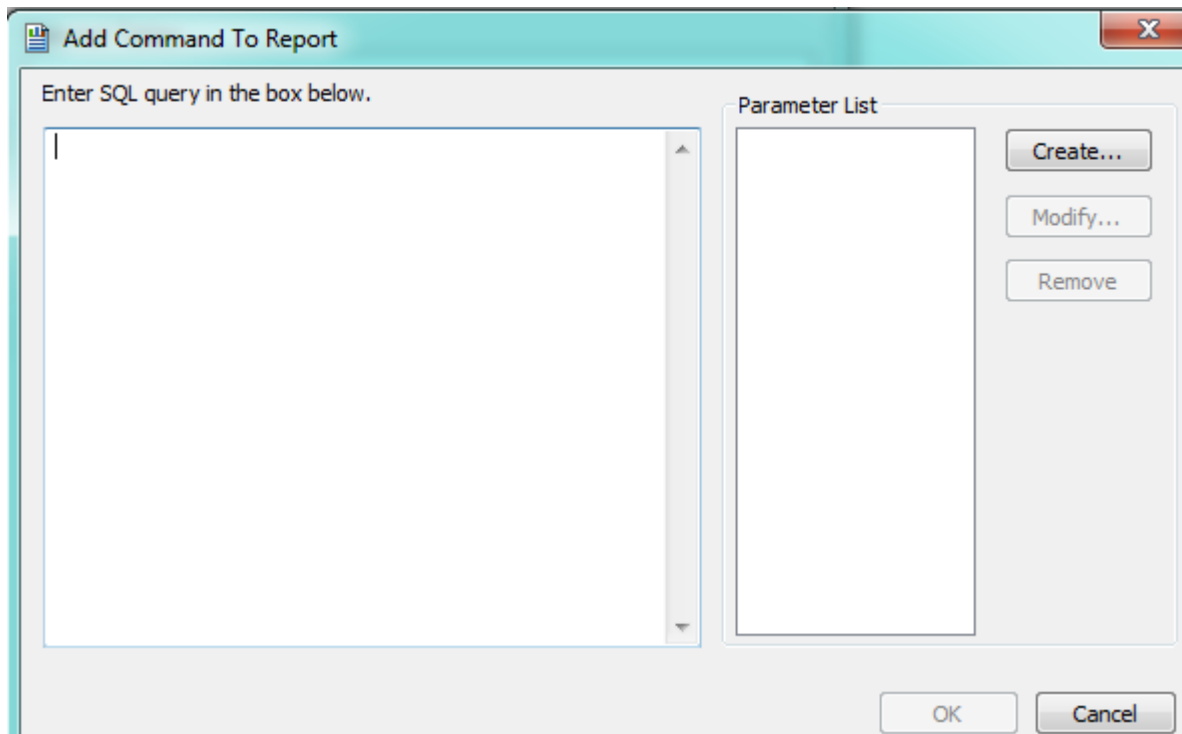
Commands do not recognize Lucity filters when the report is run.

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*.

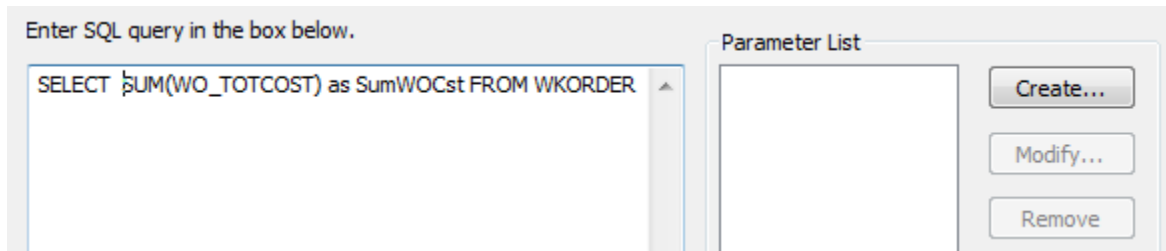
In Database Expert the Add Command would have been selected.



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 in the previous report.

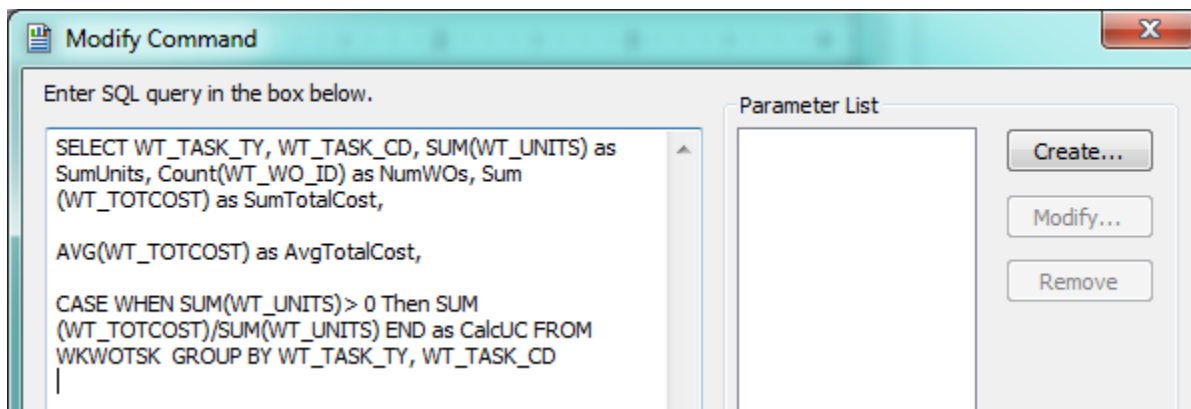
Here is another report which only used a Command.

Task Summary Report - Avg Cost/Unit Sort					7/17/2014 9:57:54PM
Task	**# WOs	Units	*Total Cost	*Avg Cost/Unit	
ENW T00 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.

Task Summary Report - Avg Cost/Unit Sort					Print Date
? Report Subtitle					Print Time
Task	**# WOs	Units	*Total Cost	Avg Cost/Unit	
D	@ Task	NumW O s	SumUnits	SumTotalCos	CalcUC
RF		Totals	d.NumWO s	d.SumTotalCos	
PF	<small> *A 'Hidden' field indicates permission to view the secured field is turned off. **Due to possible multiple tasks per work order the sum of the work orders per task will not necessarily equal the Total WOs. </small>				

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'

```

Note: _____

Image Reports

Reports using images can be helpful. There are several imaging reports used throughout the Sewer modules. At this time, Lucy Web does not support this type of report.

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

RH	@DroPath	
PH		
	TV Observation Image* Report	
	Report Subtitle	
GH1	US Structure: NT_USMAN DS Structure: NT_DSMAN Date Inspected: @DateInsp	US Address: @USAddress DS Address: @DSAddress TV Direction: TL_TVDR_TY Crew: TL_CREW
GH2	Observation #: TO_OJ Effect: TO_NOT2_TY Rating: TO_RJ	Distance: TO_FOOT Location: TO_NOT1_TY Observed GPM: TO_FLO_2
D		
GF2a	Images.rpt	
GF1a		
GF1b		
RF		
PF	Images are limited to bmp, tiff, ppx, tga, jpg, pict, png	

1. First a *Parameter field* is set up:

Edit Parameter: GBAMSDOCPATH
✕

Name: Type:

List of Values: Static Dynamic

Value Field: Description Field:

Value	Description
Click here to add item	

Options:

Option	Setting
Prompt Text	Enter GBAPath:
Prompt With Description Only	False
Default Value	
Allow custom values	True
Allow multiple values	False
Allow discrete values	True

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

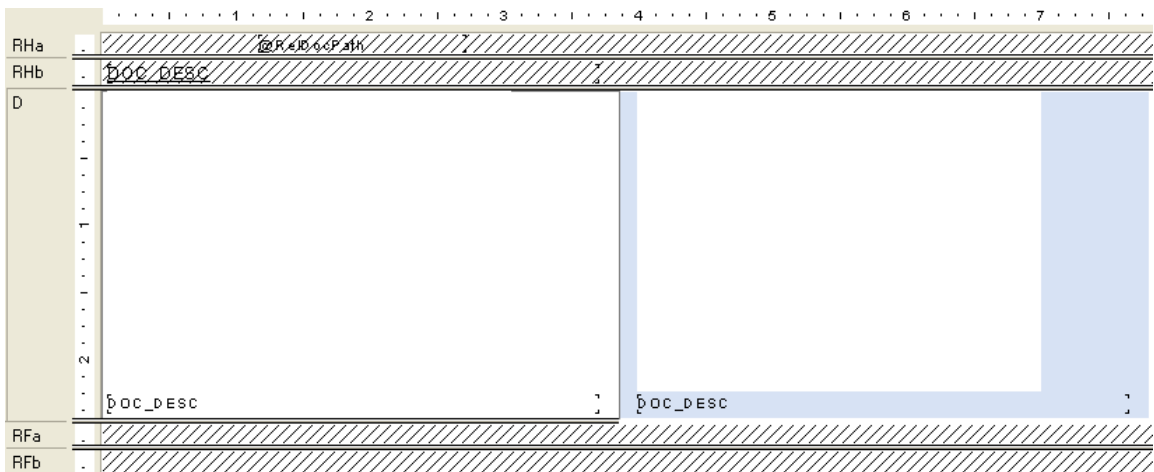
```
shared stringvar GBADocPath:={?GBAMSDOCPATH};
```

- The Image subreport is in a *Group Footer*:
- 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"
```

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

UpperCase({SWDOC.DOC_PATH}) Like ['.JPG', '*.BMP', '*.JPEG', '*.TIF', '*.TIFF', '*.TGA', '*.PNG']*

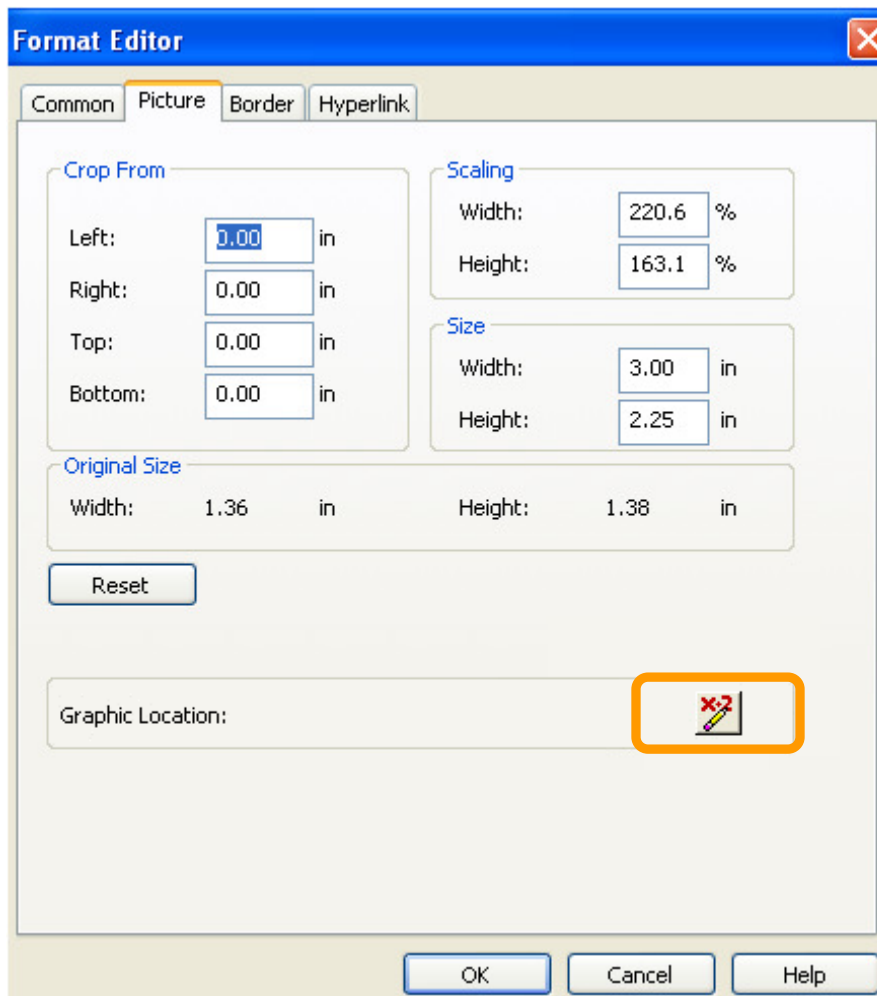


- The **RelDocPath** formula in the subreport *Header* looks like this:

```
shared Stringvar GBADocPath;
GBADocPath
```

Note: _____

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:

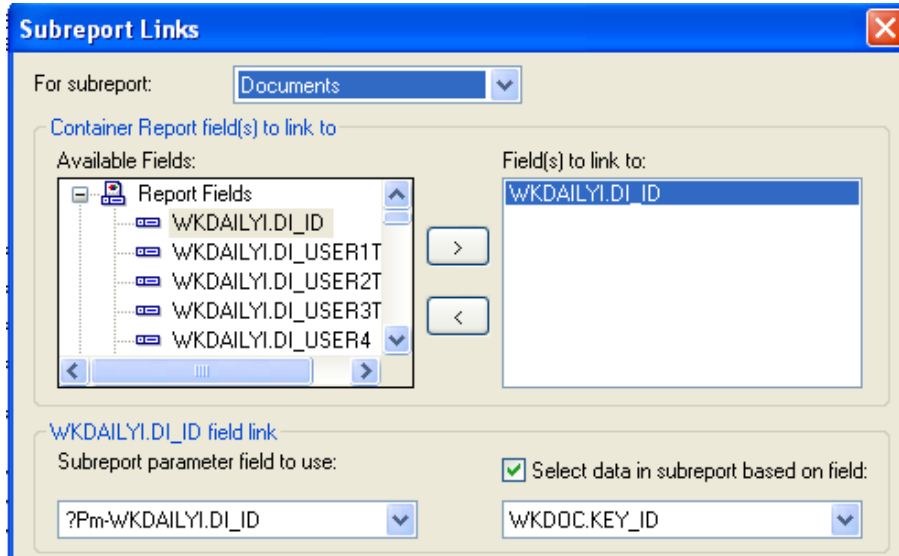

```
shared stringvar GBA DocPath;
if left ({SWDOC.DOC_PATH},13) = "$GBAMSDOCPATH"
then {@RelDocPath} + right ({SWDOC.DOC_PATH}, (Len ({SWDOC.DOC_PATH})-
13))
Else {SWDOC.DOC_PATH}
```
8. To suppress the *Detail* section when there is no image:
 - a. In *Section Expert* click on the *Details a* section.
 - b. In the Suppress (No Drill-Down) formula box there is a formula.


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

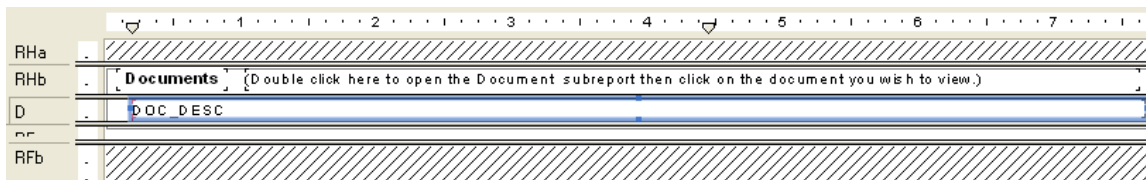
Hyperlinking

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

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



The subreport was set up like this:



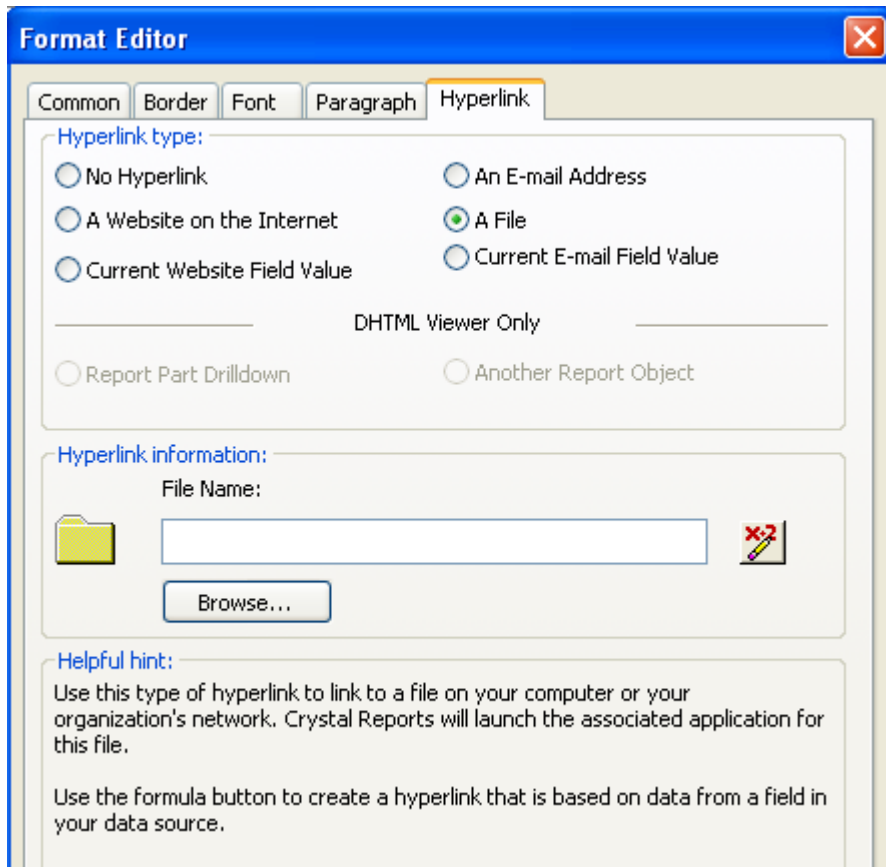
- Within the *Select Expert* the formula looked like this:

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

- For the **DOC_DESC** field you right clicked and selected *Format Field*.

Note: _____

- 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.)

Note: _____

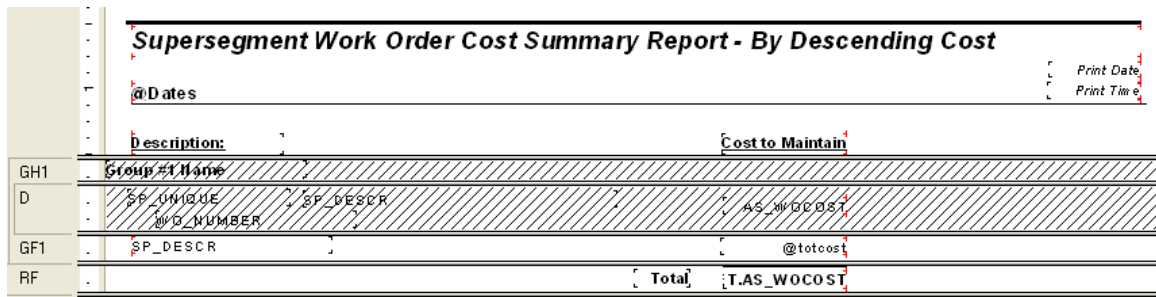
Grouping - More Advanced

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

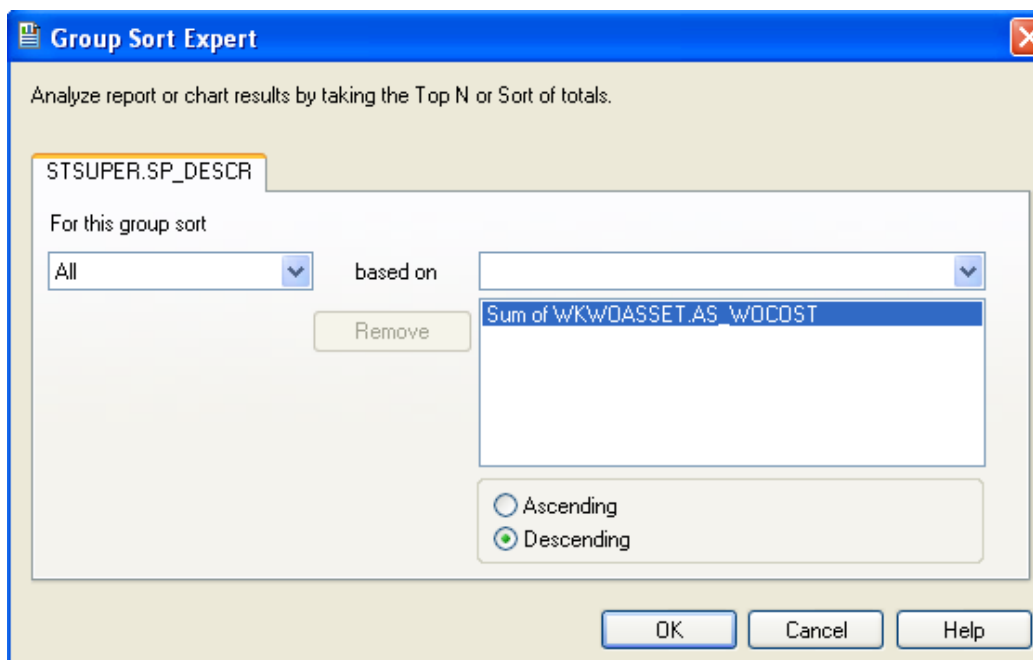
Group Sort Expert  can be used to change the field the report will sort on.

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



Supersegment Work Order Cost Summary Report - By Descending Cost		Print Date
@Dates		Print Time
Description:	Cost to Maintain	
Group #1 Name		
GH1		
D		
	SP_UNIQUE	SP_DESCR
	W/O_NUMBER	AS_WOCOST
GF1	SP_DESCR	@totoest
RF	Total	T.AS_WOCOST

- 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.



Group Sort Expert

Analyze report or chart results by taking the Top N or Sort of totals.

STSUPER.SP_DESCR

For this group sort

All based on

Remove

Sum of WKW/OASSET.AS_WOCOST


Ascending
 Descending

OK Cancel Help

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.

GH2a	<p>The following streets are designated as <u>one-way streets</u> pursuant to authority granted in Section 35-303:</p> <p>Regulation Number Location Restriction</p>
GH2b	<p>Trucks over 12,000 lbs. shall be prohibited on the following streets pursuant to authority granted in Section 35-782:</p> <p>Regulation Number Location</p>
GH2c	<p>The following streets are designated for a <u>maximum speed limit</u> as indicated, and for day and/or night as indicated, pursuant to authority granted in Section 35-267:</p> <p>Regulation Number Location Restriction</p>

- 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.

GH1a	<p>Employee Usage Detail</p> <p>Employee Number Employee Name Labor Hrs Labor Cost OT Hrs OT Cost Total Cost</p>
GH1b	<p>Equipment Usage Detail</p> <p>Equipment Number Equipment Name Hours Used Total Cost</p>
GH1c	<p>Material Usage Detail</p> <p>Material Number Material Name Unit of Measure Qty Used Total Cost</p>
GH2	<p>Group #2 Name</p>
D	<p>@LbHrs @LbCost UNITS R_TCST OTHR_TCST OT WR_UNITS WR_UNIT_C WR_TCST ST</p>
GF2a	<p>WR_RSRC_CD WR_RSRC_TY #LbHrs #LbCost #OTHrs #OTCost #EmTotCost</p>
GF2b	<p>WR_RSRC_CD WR_RSRC_TY #HrsUsed #Cost</p>
GF2c	<p>WR_RSRC_CD WR_RSRC_TY WR_UOM_TY #Qty #MatCost</p>
GF1a	<p>Totals: #TotLbHrs #TotLbCost #TotOTHrs #TotOTCost #GrTotCost</p>
GF1b	<p>Totals: #TotHrs #EqTotCost</p>
GF1c	<p>Totals: #TotQty #MtTotCost</p>
RF	<p>The referenced date is the Resource End Date.</p>

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]

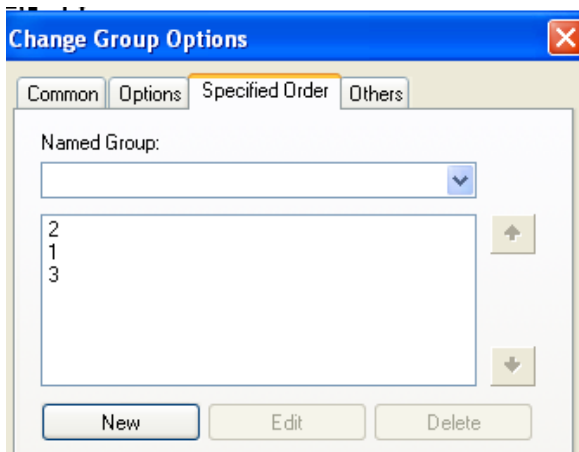
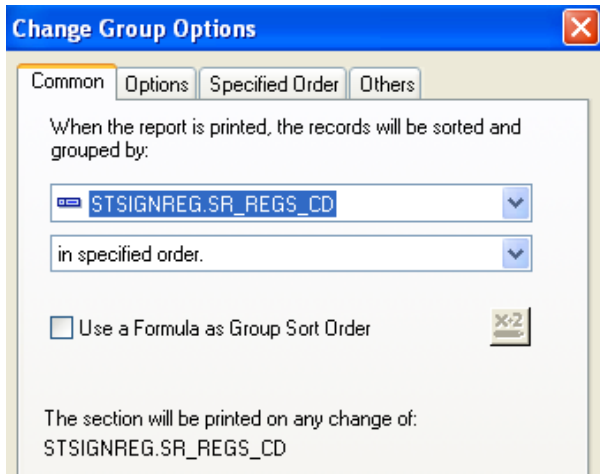
Specific Order of Grouping

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

PH	-	TRAFFIC REGULATIONS -	SR_REG_TY		
	-	?Post Date			
	-	@Page			
GH1	-	Group #1 Name			
GH2a	-	The following streets are designated as <u>one-way streets</u> pursuant to authority granted in Section 35-303:			
	-	Regulation	Location	Restriction	
	-	Number			
GH2b	-	Trucks over 12,000 lbs. shall be prohibited on the following streets pursuant to authority granted in Section 35-782:			
	-	Regulation	Location		
	-	Number			
GH2c	-	The following streets are designated for a <u>maximum speed limit</u> as indicated, and for day and/or night as indicated, pursuant to authority granted in Section 35-267:			
	-	Regulation	Location	Restriction	
	-	Number			
D	-	SR_UNIQUE	RedLoc.rot	Restrictions.rot	
GF2	-				
GF1	-				

Note: _____

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

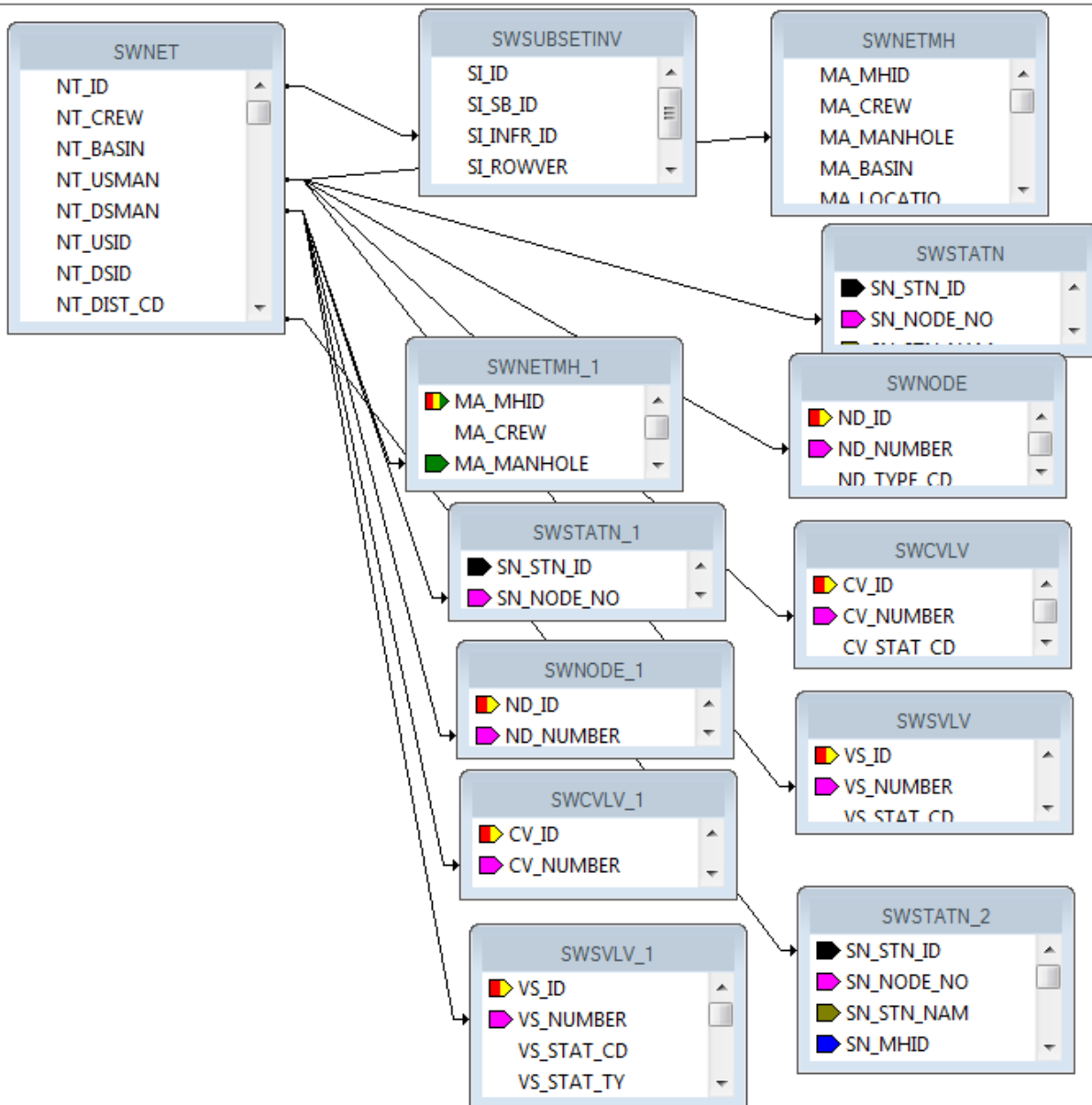


Duplicate Table Use

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) - 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,"

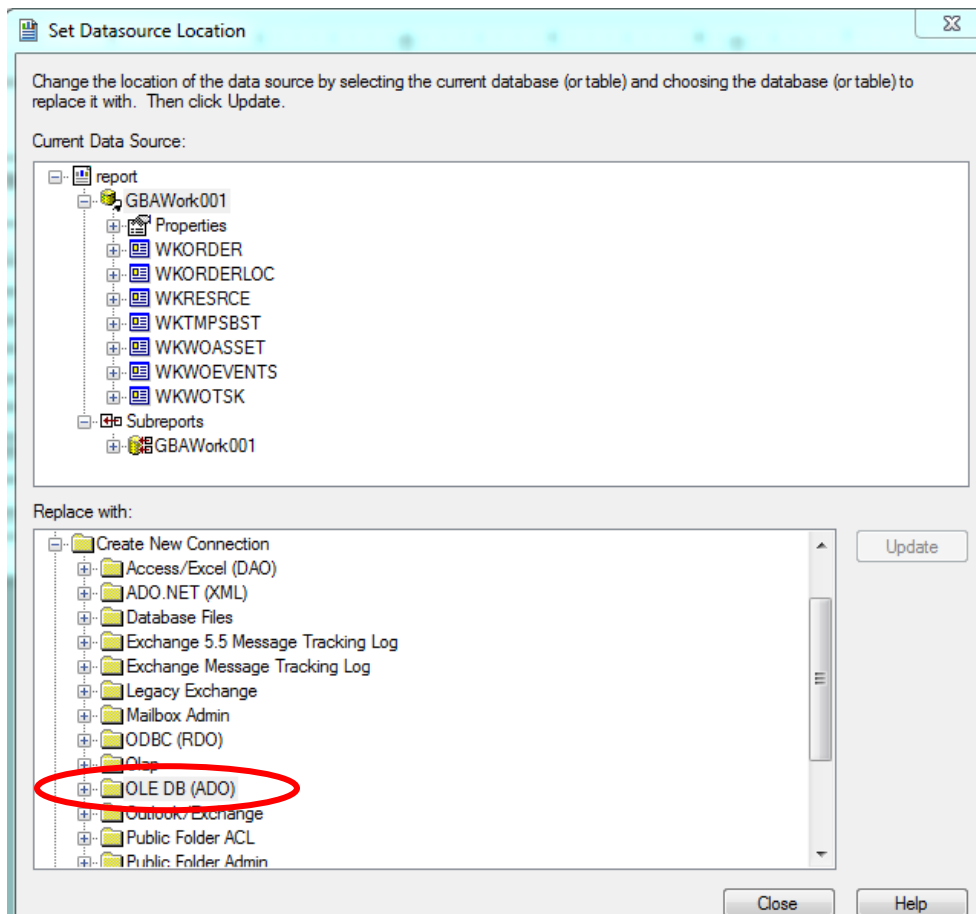
Note: _____

Web Reports

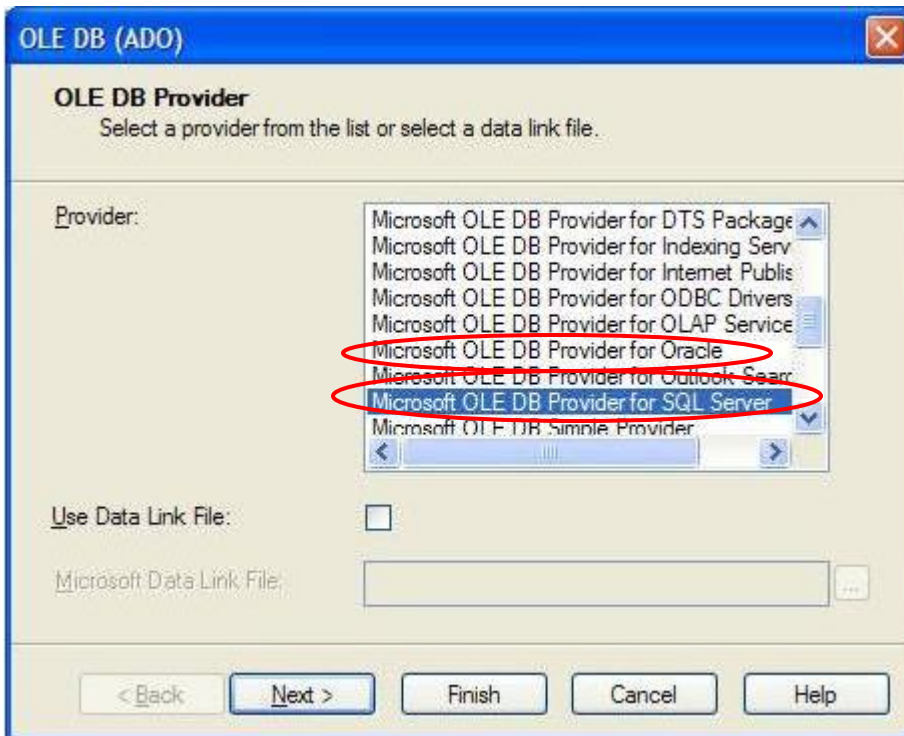
Converting Desktop Reports to use in the Web

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

1. Open the report in Crystal Reports.
2. Go to Database>>Set Datasource Location.
3. Within the lower box (*Replace with:*) open *Create New Connection*.
4. Select *OLE DB (ADO)*

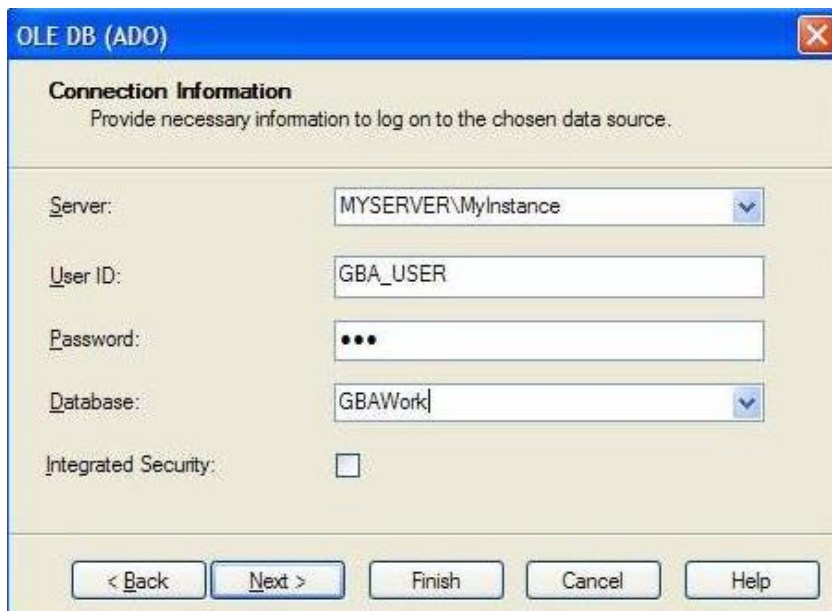


5. You will then be prompted to select a provider. If you are using SQL Server use the *Microsoft OLE DB Provider for SQL Server*. For Oracle use *Microsoft OLE DB Provider for Oracle*.



6. Next, enter the Connection information.
7. For SQL Server, include the SQL Server Name, User ID, Password, and Database. Your Lucy Administrator should be able to provide this information. The Database is specific to the report database that it was created with such as GBAWork or GBAEquip.

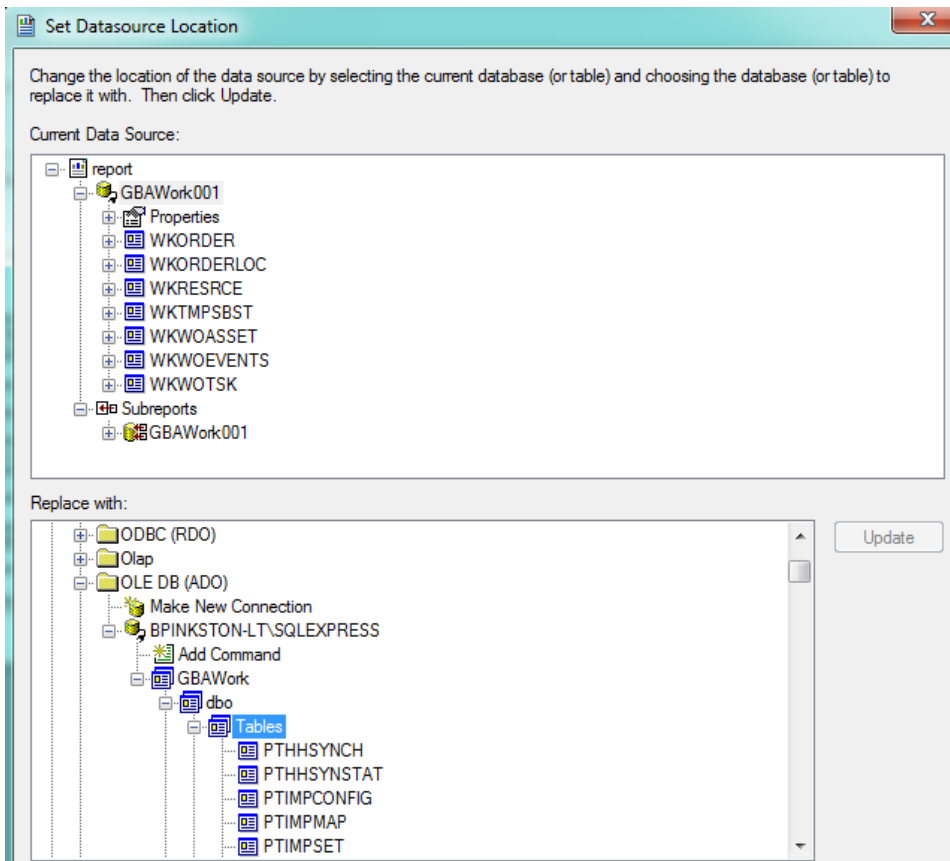
Note: For 2015r2 post data unification it will be one database. The name will be whatever you have assigned the database.



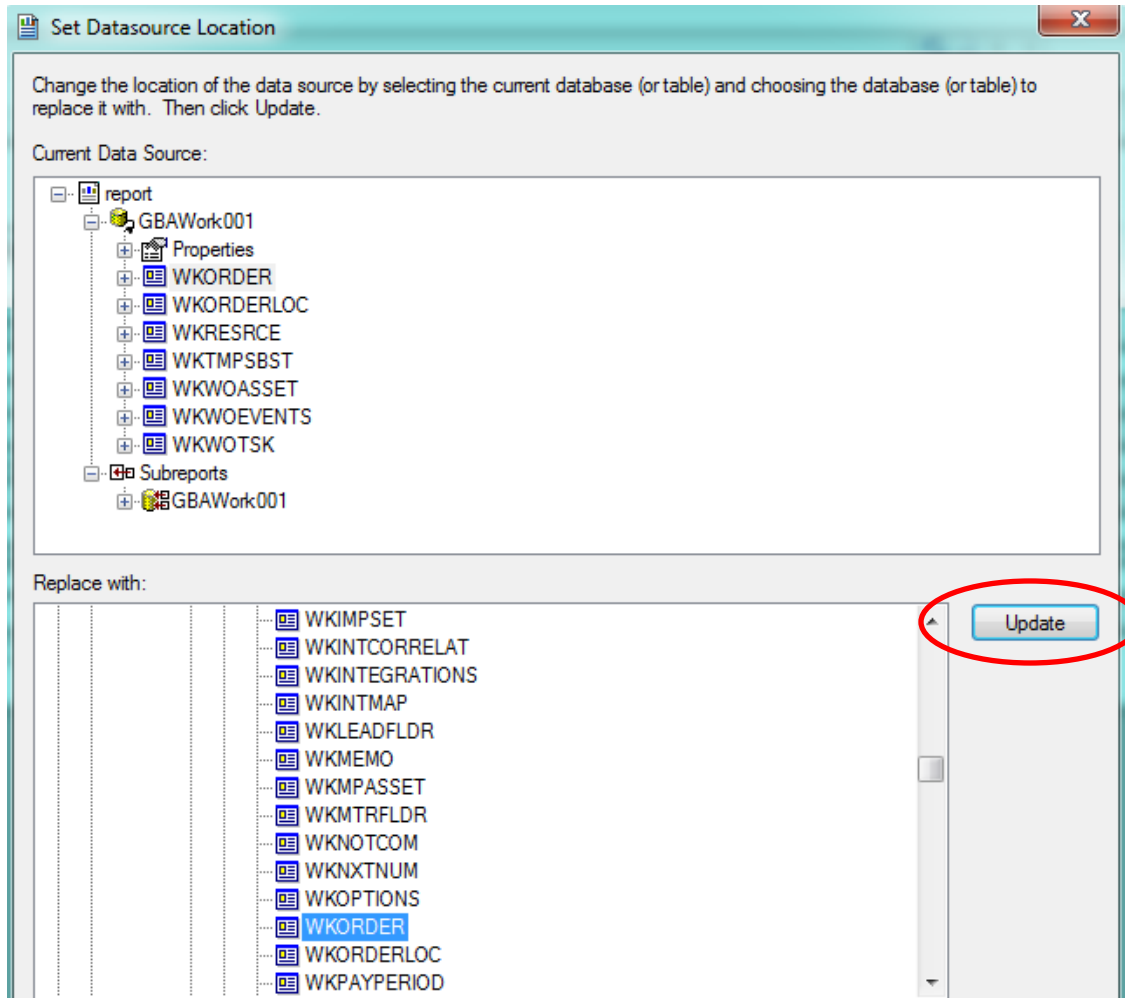
- For Oracle, include the Service, User ID, and Password.



8. 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!



- To do this, click on a table in the top box (*Current Data Source*) and the same table in the bottom box (*Replace with*) and then click on *Update*. Do this for each table, even the ones listed under Subreports. Note: *Properties* is not a table to be updated.



After the report is converted for Web use it will need to be added to the appropriate report folder.

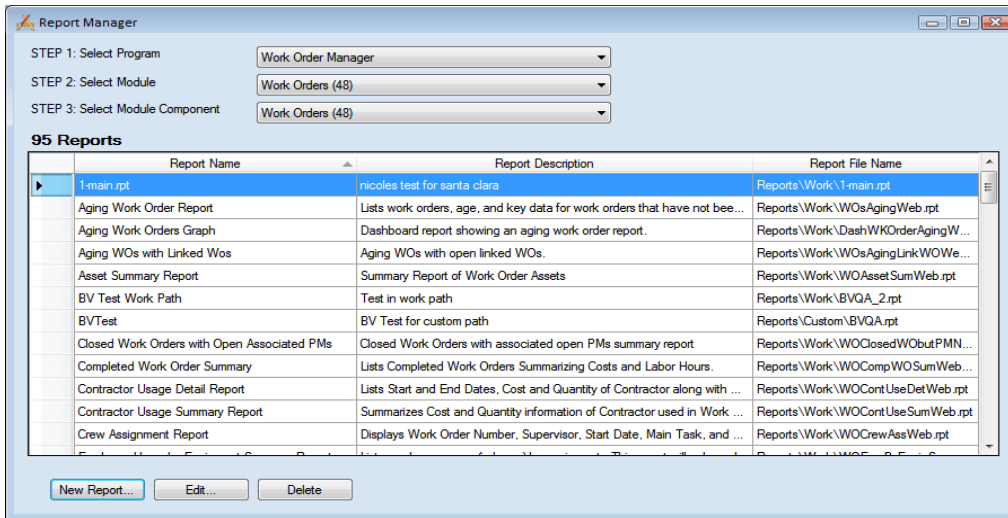
Adding Custom Web Reports (Prior to Version 2015)

The converted report then needs to be added to a report folder, typically found at **intpub\wwwroot\LucityWeb\Reports**. Drop the report in the correct folder such as Work or Equip. Depending on your system and number of custom reports, you may have a separate Custom Reports folder and within this there may be folders for the type of reports such as Work or Sewer.

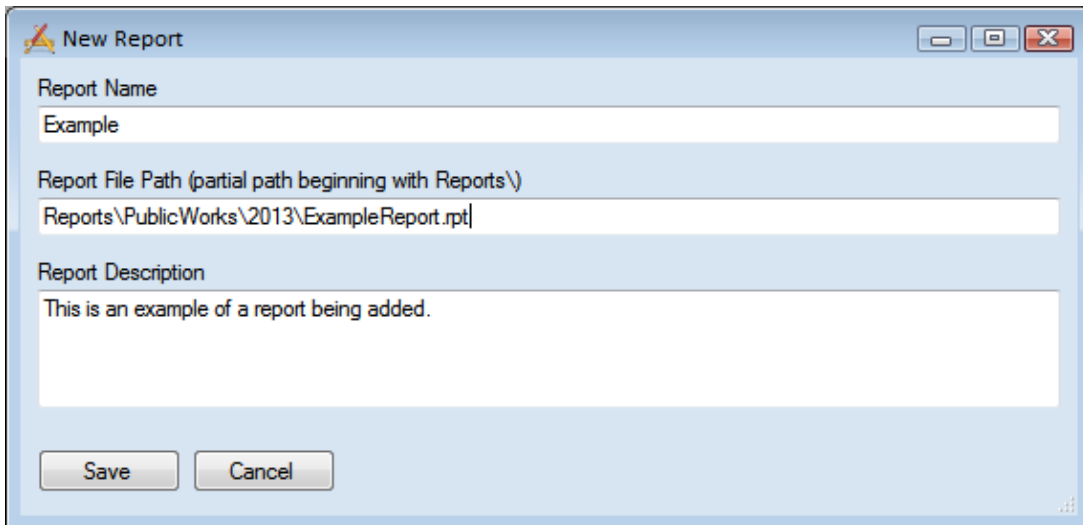
Then the report will need to be added to the list of available reports in the module the report is to be run from. There are two different methods for adding reports in the web.

Method 1 - Through the Lucy Administration Tool

1. Launch the Lucy 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:

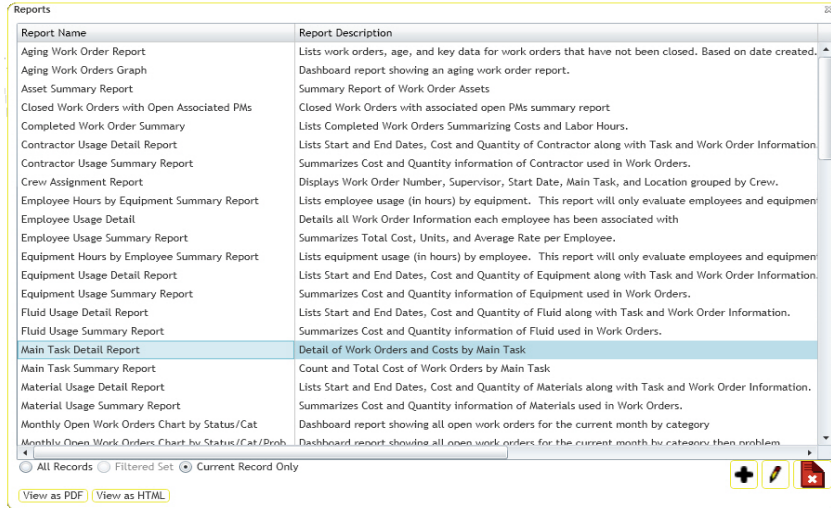


4. Enter the **Report Name**.
5. Enter the **Report File Path**. This should always start with **Reports** and then the rest of the path to the report including the **.rpt** file extension.
6. Enter the **Report Description**.
7. Click **Save**. The new report will be added to the Report Manager.

Method 2 - Through Lucidity Web

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

To launch the Reports tool click the  on the toolbar. The following window will appear:



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

Add Report

Report Name:

Report Description:

Report File Path (partial path beginning with Reports\):

Standard Report
 Quick Report - All Records
 Quick Report - Filtered Set
 Quick Report - Current Record Only

Edit Report

Report Name:

Report Description:

Report File Path (partial path beginning with Reports\):

Standard Report
 Quick Report - All Records
 Quick Report - Filtered Set
 Quick Report - Current Record Only

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**.

Custom Web Reports Version 2015 (sp 1) and Beyond

The Lucity software is moving the web design to be used as Software As A Service (SAAS). To accommodate this direction, the Custom Web reports are handled in a different way.

There are two locations that Lucity will now use to manage custom reports.

- One is a new folder that can be seen in **inetpub > wwwroot > LucityWeb** that is called **ReportsCustom**. The client should never be in this folder.
- The other is a new location that the client sets up. (**Report Share**) Once this is set up and the existing reports moved into this new folder then this folder should not be directly used by the client unless copying a report to be modified. Indirectly, the new custom reports will be uploaded to this location.

Note: _____

Understanding the Process

How Lucy manages Custom Reports

Lucy makes it easy for an agency to manage its custom reports by managing the file locations for them and providing an easy user interface for uploading report files. To make this process work there are several components on the back end. The following table describes those components and the diagrams show how they work together to complete a process.

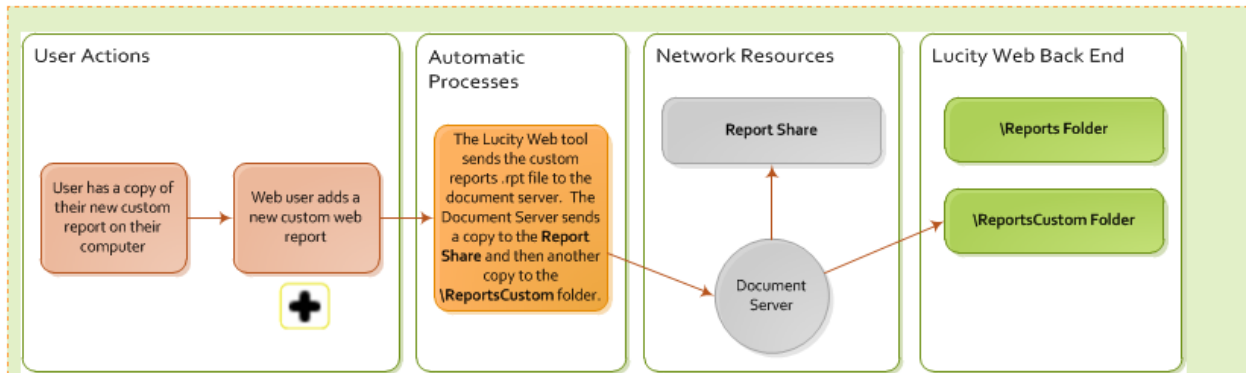
Components

Lucy Web Reports Tool	A tool that enables Lucy Web users to add, update and run Lucy reports.
Document Server	A Lucy Web Server application that enables users to move files (including reports) around using Lucy Web .
Report Share	The folder that stores a user-accessible copy of each custom report's .rpt file. <ul style="list-style-type: none">• The path is set in the Lucy Administration Tool > System > Settings > Documents > Path where reports are stored setting.• Lucy Web does not run reports from this location.• Users can copy reports out of this folder to modify them.• The reports in this folder are copied to the /ReportsCustom folder daily.
/Reports Folder	The folder that stores a Lucy Web -accessible copy of each standard report's .rpt file. <ul style="list-style-type: none">• Lucy Web runs all of Lucy's built-in, standard reports from this location.• Historically, custom reports were added to this folder and they were mixed in with standard Lucy reports.• Users can copy reports out of this folder to make custom reports.• The reports in this folder should not be modified.
/ReportsCustom Folder	The folder that stores a Lucy Web -accessible copy of each custom report's .rpt file. <ul style="list-style-type: none">• Lucy Web runs all custom reports from this location.• The reports in this folder should not be modified.• The reports in this folder are overwritten with copies from the Report Share folder.

Processes

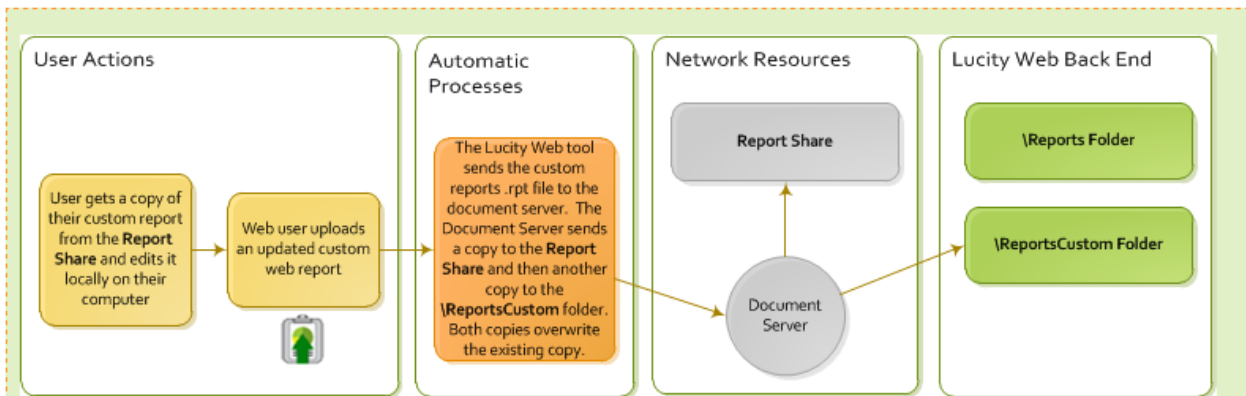
These are the processes that occur in the background when users are working with **Lucity Web** and custom reports.

▼ User adds a custom report

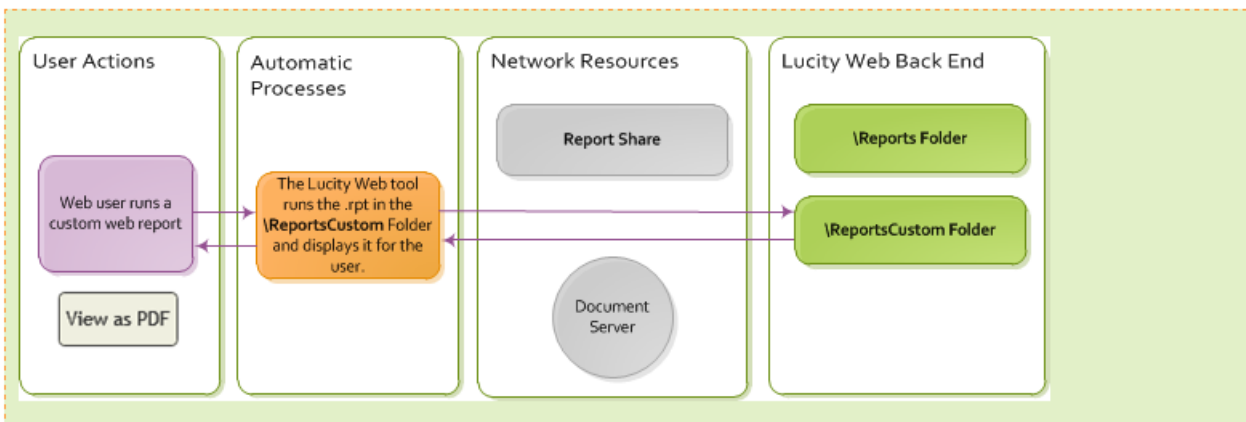


Note: Give this process a minute to complete before attempting to run the new report from Lucity.

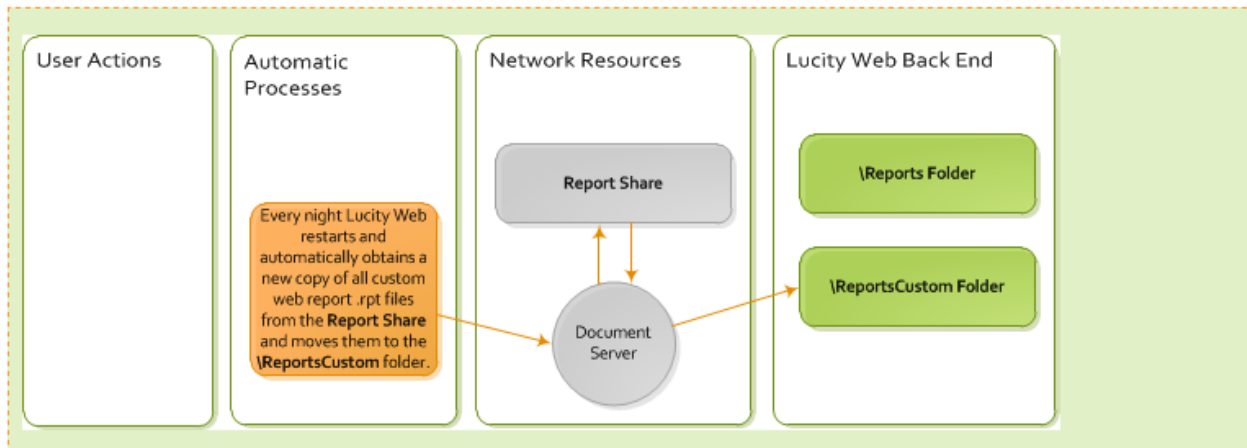
▼ User updates a custom report



▼ User runs a custom report



▼ Automatic custom report processes



Best Practice for Making the Change

The existing custom reports were supposed to be working with the upgrade but did not quite make this transition correctly. There have also been some issues relocating the established custom reports and updating the new path. Support has established the best method for handling custom web reports is to add them into the system as if they were new reports.

Setup

1. If the client doesn't currently have a location outside of Lucy to store and work on custom reports then this should be established first.

It might be a good idea to have one shared file for all custom reports for your company. This is separate from Lucy!

All existing custom reports currently in `wwwroot\LucyWeb\Reports` should be placed in this new folder or subfolders in this location.

Example folder name: **CompanyCustomWebReports**.

2. A second location needs to be established which Lucy will use. This needs to be a folder that the WebServer (IIS) can get to. Ideally, this would not be on the web server. That means it will be on a network share. Something like [\\<Servername>\Shared\CustomWebReports](#).

Example: [\\ClientServ-01\WaterGroup\CustomWebRpts15](#)

Where "[\\ClientServ-01](#)" references the server, "WaterGroup" the Shared location and "CustomWebRpts15" the folder name for the custom web reports.

3. This path then needs to be entered in:

Lucy Administration Tools > System > Settings... > Documents tab > Path where Reports are stored. Enter the new path for the Custom Web reports.

(*Example:* [\\ClientServ-01\WaterGroup\CustomWebRpts15](#))

Save

4. Clear Caches
 Lucy Administration Tools > System > Clear All Site Caches... > Clear All Caches
 button.

5. An agency may wish to create a directory structure in this location to separate reports by system; however, this is optional:
 \\<Servername>\Shared\CustomWebReports\Work\ <reports for the work program go here>
 \\<Servername>\Shared\CustomWebReports\Sewer\ <reports for the sewer program go here>



6. Once the custom report path has been established and the custom reports are in the new location outside of Lucy (example: **CompanyCustomWebReports**), then the reports should be added back into Lucy.

7. Within the module the custom web report originally ran from, delete the report and add as a new report.

How To Add New Custom Reports



Going forward, when adding new custom reports, upload the report using *Lucy Web*. The system will place the report in the report-share automatically.

1. In *Lucy Web*, open the module you would like to add the report to.
2. Click the Reports  button on the toolbar. The *Report* tool opens.
3. To add a new report, click the Add button . The following dialog appears:

V2015

Add Report

Report Name:

Report Description:

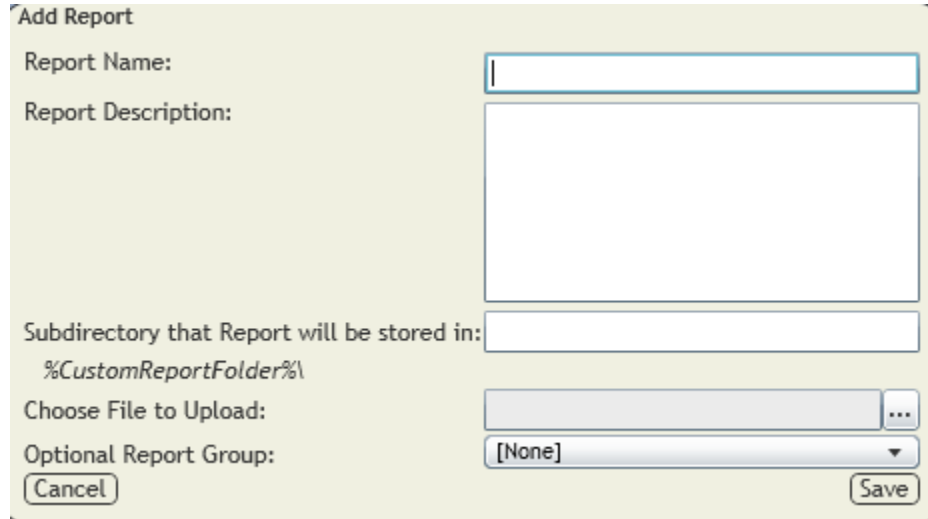
Report File Path:

Choose File to Upload:

Report Group (optional): ▼

Standard Report
 Quick Report - All Records
 Quick Report - Filtered Set
 Quick Report - Current Record Only


V2015r2



4. Enter the *Report Name* and *Report Description*.
5. In the *Report File Path*, enter the name of the subfolder to which you want to upload the report file (ex. Work, Sewer) or leave blank.

Note: Reports are automatically uploaded to the location specified in the Lucity Administration Tool's "Path where Reports are stored" setting, which may be found in the *System > Settings > Documents* tab.

Note: If the folder name entered does not exist, the system will create a folder with that name within the folder to which the reports are uploaded.

6. Under *Choose File to Upload*, click the Browse button  and select the .rpt file you would like to upload to the web server. The report would ideally be located in the folder established in Step 1 at the top of this document (**CompanyCustomWebReports**) or a subfolder within, however, it can be brought in from any location. This will put the report into the location created for Custom Web reports that was designated in the *System Settings Report File Path*.

Note: The name of the .rpt file being uploaded cannot match any other .rpt files.

7. Assign the report to a *Group* (optional).
8. Choose or change the *Type* of report (V2015):
 - **Standard Report** - The report is generated when the user clicks the **Report** tool, selects the report and chooses how to run it.
 - **Quick Report - All Records** - Adds the report to the *Quick Report* drop-down list, located next to the **Report** tool. The report always runs against all records.
 - **Quick Report - Filtered Set** - Adds the report to the *Quick Report* drop-down list, located next to the **Report** tool. The report always runs against the current filter set.
 - **Quick Report - Current Record Only** - Adds the report to the *Quick Report* drop-down list, located next to the **Report** tool. The report always runs against the current record.

Add Report

Report Name:

Report Description:

Report File Path:

Choose File to Upload: ...

Report Group (optional): ▼

Standard Report
 Quick Report - All Records
 Quick Report - Filtered Set
 Quick Report - Current Record Only

For V2015r2 these options are available in the Edit mode.

9. Click *Save*.

Editing Reports





Editing allows changes in the Report Name, Description, Group or Standard/Quick Report options. **Update** is used to save the changes.

Updating Custom Reports



When existing Custom reports need to be revised, this should be done outside of the Lucity system. The revised reports could be worked on or loaded into the location established for your company, mentioned in step 1 (Example: **CompanyCustomWebReports**).


1. Open *Lucity Web* and go to the module associated with the report.
2. Click the Reports button  on the toolbar. The *Report* tool opens.
3. Select the report to update from the list and click the Update Report File button .

Update Report File (StaticWOSumWeb.rpt.rpt)

This will replace the existing .rpt on the server.
Please ensure that the file selected has the same name as the existing file.

...

Note: The file name has to be identical to the file being replaced.

4. Click the Browse button  and select your updated .rpt file from the location of the revised report (Example: **CompanyCustomWebReports**).
5. Click **Update**.
6. The new copy of the report will be uploaded and will replace the existing report file in the Lucy system (the location established in the System Settings Report Path).
7. Wait a minute before trying to run the updated version of the report.

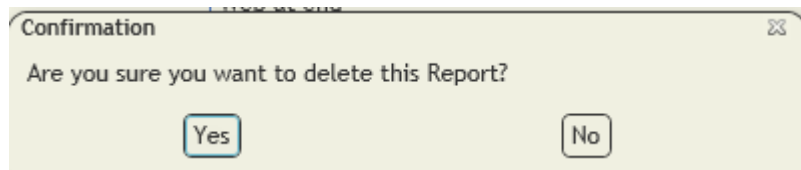
Note: The reports that Lucy provides with the software (standard reports) cannot be updated.

Delete Report



For version 2015 sp0, if the User's Report Security isn't set up properly, a user can delete the entire report file. This caused problems with several clients so this was changed.

In 2015 sp1 the Delete option queries if the user would like to delete the report.



If you select Yes, the report is removed from the dialog but the report remains out in the Custom Report folder as an orphan. The IT administrator can retrieve it if necessary or delete it.

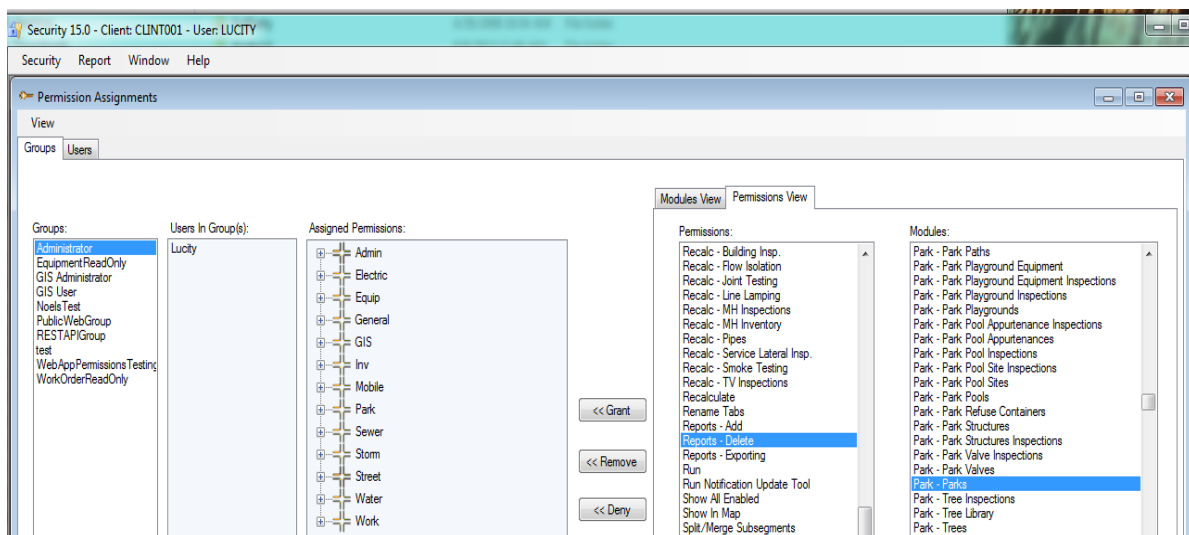
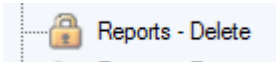
The plan for version 2016 is to be able to run a report that would show the Report orphans. There will possibly also be a way of deleting the report file (.rpt) from the report dialog thus eliminating the orphan file all together.

Note: _____

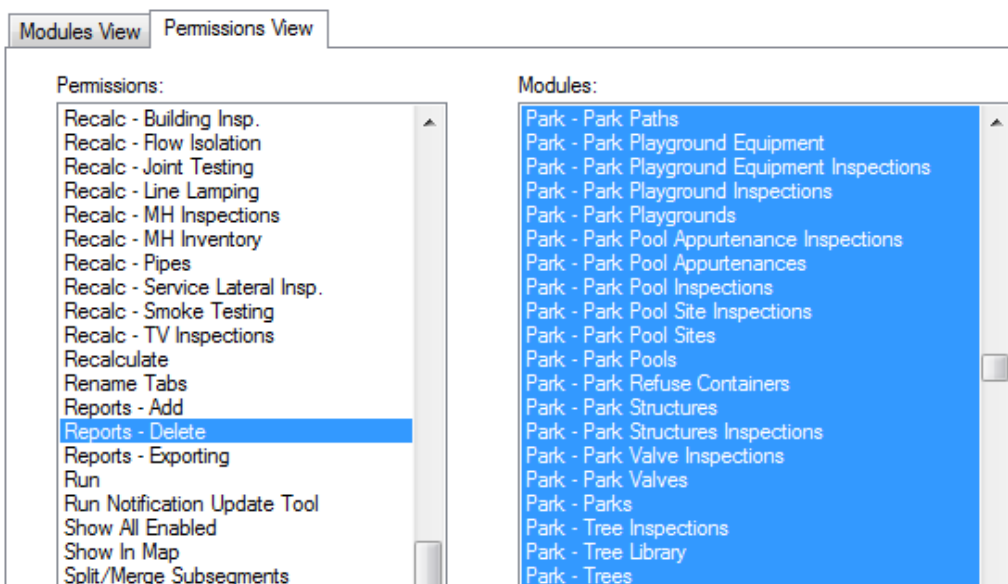
Report Security

The Report Delete permission can be denied.

1. In **Lucity Security > Permissions Setup**
2. In either the **Groups** and/or **Users** tab, Select the Group.
3. On the right side **Permissions View**, **Permissions** box, click on **Reports - Delete**.
4. In the **Modules** box select individual modules or multiples.
5. In the center select the <<**Deny** button.
6. In the **Assigned Permissions** box you can check the module's **Reports - Delete** permission.



Larger view of right side with multiple modules selected.

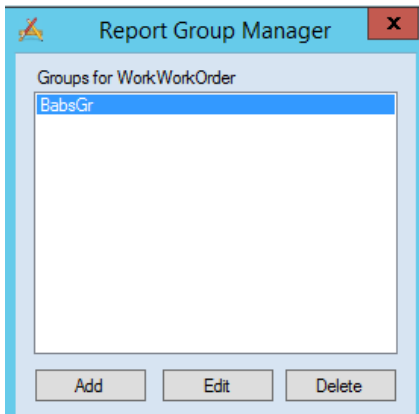


Web Report Grouping

In Lucity 2014R2 there is a new option for Report Grouping.

1. In Lucity Administration Tools go into Reports > Report Manager
 - a. Select Program
 - b. Select Module
 - c. Select Module Component
2. Click on Report Groups

The Report Group Manager window will show.



- Select *Add* to add a group.
 - Type in a Group Name
 - Save
 - Select *Edit* to edit an existing group.
 - Type in a revised Group Name
 - Save
 - Select *Delete* to delete an existing group.
3. Close the Report Group Manager window.
4. In the Report Manager select a report to include in a group.
5. Click *Edit...*
6. In the Update Report window there is an option for *Report Group (optional)* with a drop down arrow. Select the group the report is to be associated with.
7. Save
The Group Name should now show in the Report Group column for the selected report.
8. Repeat from step 4 for all reports to be associated with the group.
9. When the Web module's report list is opened, the top left corner now has a drop down box for Report Group. When the group is selected, only the reports assigned to the group will show.

If the newly added group does not show in the list then close Lucity and go into Lucity Administration Tools > System > Manage Web Site Caches...

- Scroll to the bottom and click **Clear all Caches.**
- Close
- Reopen the Web module's report list and the Report group should now be present in the Report Group drop down box.

Web Reports and Parameters

Typically record selection occurs with a filter in Lucity, 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.

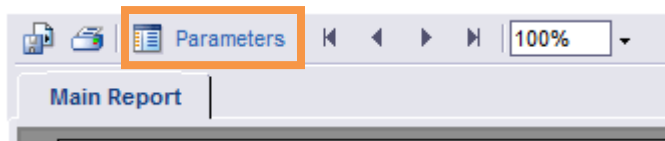
Parameters set up with a Static list to select from will allow selection of a single value to run the report in either PDF or HTML view.

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 view.

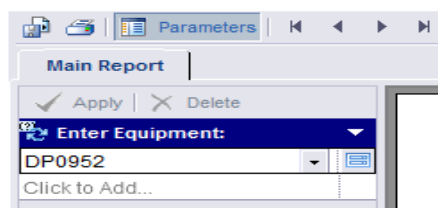
- A parameter that is set up to allow multiple values will only accept a single value when the report is run.
- The above limitations apply to web reports Viewed as PDF or developed in Crystal XI or earlier. Reports revised or created in Crystal 2008 can be set up to allow the parameter to be queried 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

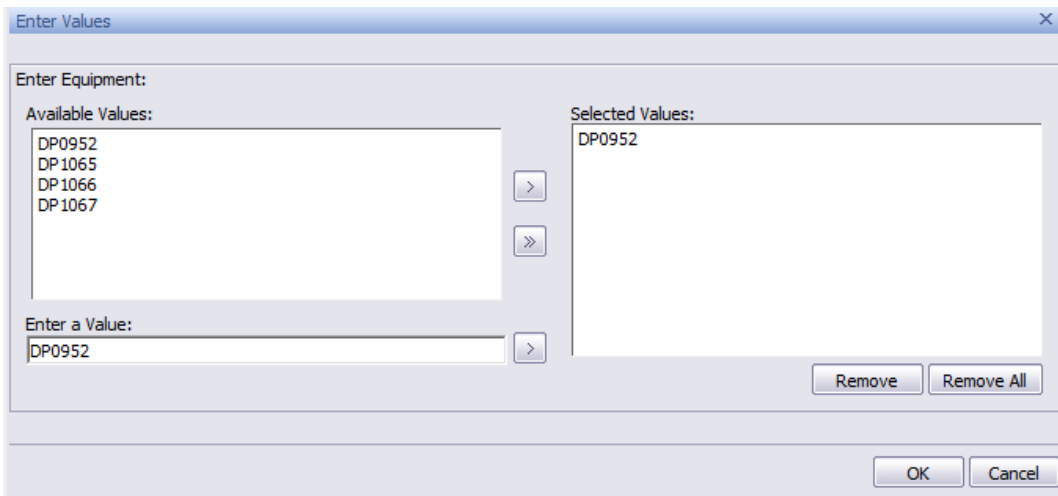
When this setting is **Editable** and the Web report opened with the HTML view, the **Parameter** button will be active for use to modify the Parameter selection. To use this option, complete the following steps:



1. 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.

Note: _____

Dynamic Parameters

Starting in v2015, the Dynamic Parameter pick list is available when running the report in Lucity Web. The way to achieve this is by naming the parameter the same as the field name.

For example, if the Work Category field was to be set up as a dynamic pick list, the Name would be entered as WKORDER.WO_CAT_TY. The Prompt Text usually defaults to the Name field (Enter WKORDER.WO_CAT_TY) so this would likely be modified to “Enter Category”.

Dialog box: Edit Parameter: WKORDER.WO_CAT_TY

Instructions: Edit a parameter and list of values.

Name: WKORDER.WO_CAT_TY (circled in red) | Type: String | List of Values: Dynamic

Prompt Group Text: [Empty]

Choose a Data Source: New Existing | WO_CAT_TY - Prompt Group

Value	Description	Parameters
WO_CAT_TY	(None)	[?] WKORDER.WO_CAT_TY

Value Options - Level 1 (WO_CAT_TY):

Option	Setting
Show on (Viewer) Panel	Editable
Prompt Text	Enter Category:
Sort Order	Ascending by Value
Prompt With Description Only	False
Optional Prompt	False
Allow multiple values	True

Buttons: OK, Cancel, Help

Web Filters

Date

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'

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.

Adding Reports to Dashboard (prior to V2015r2)

Standard Reports

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

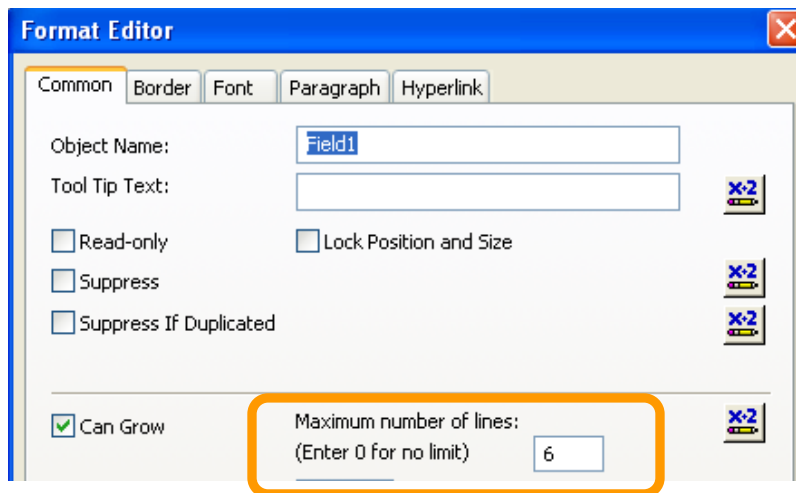
1. In *Lucity Admin Tools* click on **Dashboard > Dashboard/Preferences**.
2. If **Lucity** is in the list of users on the left, select it and click **Edit**. If it is not there, click **Add**.
 - o If you have to **Add**, then on the next screen select **Lucity**.
3. On the **Dashboard tabs** screen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

Odd Things Happen

Field not showing all of the text

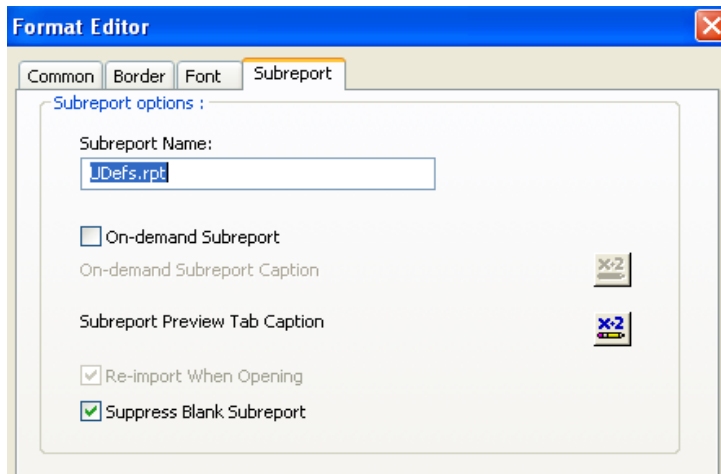
Check the Field Format Editor in the common tab to make sure the *Can Grow* option is checked. If it is checked and the field is still cutting off the data then make sure the maximum lines is set at "0".

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.



Web (PDF) Page Number Limitation

Prior to V2015r2

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.

V2015r2

PDF (Basic View) page number can be modified in **Lucity Administration > System > Settings > Web Performance** tab > **Max # of Pages on PDF Crystal Reports** (to reduce performance impact on server)

Lucity Administration 15.5 - Client: Lucity Eval Data - User: LUCITY

System Navigation Forms Reports GIS Security Windows Help

System Settings

Appearance	Designer Automation	Documents	Email	General	GIS 3rd Party Integrations	GIS Desktop	GIS Edit Integration	GIS Routing	GIS Web
Mobile	REST API	SaaS	Security	Settings with custom interface	TimeSheet	Web Performance	Web Site	Work	Work Zones
Description		Value							
Allow Animations to run inside of the website.		TRUE							
Enable Column Resize for grid pages with less than this number of rows		50							
Exclude Desktop Records in Add Mode (When TRUE, queries will be slower)		FALSE							
Favor configuration over performance for business files		FALSE							
Max # of Pages on PDF Crystal Reports (to reduce performance impact on server)		20							
Persist changes to page size in List View		TRUE							
The number of minutes until a dashboard report should be refreshed.		240							

Note: _____
