

Intergraph Smart 3D™ Reports Customization Course Guide



PROCESS, POWER & MARINE

Version 2014

November 2014

DSP3D-TP-100087A



INTERGRAPH®

Copyright

Copyright © 2014 Intergraph® Corporation. All Rights Reserved. Intergraph is part of **Hexagon**.

Including software, file formats, and audiovisual displays; may be used pursuant to applicable software license agreement; contains confidential and proprietary information of Intergraph and/or third parties which is protected by copyright law, trade secret law, and international treaty, and may not be provided or otherwise made available without proper authorization from Intergraph Corporation.

Portions of this software are owned by Spatial Corp. © 1986-2014. All Rights Reserved.

Portions of the user interface copyright 2012-2014 Telerik AD.

U.S. Government Restricted Rights Legend

Use, duplication, or disclosure by the government is subject to restrictions as set forth below. For civilian agencies: This was developed at private expense and is "restricted computer software" submitted with restricted rights in accordance with subparagraphs (a) through (d) of the Commercial Computer Software - Restricted Rights clause at 52.227-19 of the Federal Acquisition Regulations ("FAR") and its successors, and is unpublished and all rights are reserved under the copyright laws of the United States. For units of the Department of Defense ("DoD"): This is "commercial computer software" as defined at DFARS 252.227-7014 and the rights of the Government are as specified at DFARS 227.7202-3.

Unpublished - rights reserved under the copyright laws of the United States.

Intergraph Corporation
300 Intergraph Way
Huntsville, AL 35813

Documentation

Documentation shall mean, whether in electronic or printed form, User's Guides, Installation Guides, Reference Guides, Administrator's Guides, Customization Guides, Programmer's Guides, Configuration Guides and Help Guides delivered with a particular software product.

Other Documentation

Other Documentation shall mean, whether in electronic or printed form and delivered with software or on eCustomer, SharePoint, or box.net, any documentation related to work processes, workflows, and best practices that is provided by Intergraph as guidance for using a software product.

Terms of Use

- a. Use of a software product and Documentation is subject to the End User License Agreement ("EULA") delivered with the software product unless the Licensee has a valid signed license for this software product with Intergraph Corporation. If the Licensee has a valid signed license for this software product with Intergraph Corporation, the valid signed license shall take precedence and govern the use of this software product and Documentation. Subject to the terms contained within the applicable license agreement, Intergraph Corporation gives Licensee permission to print a reasonable number of copies of the Documentation as defined in the applicable license agreement and delivered with the software product for Licensee's internal, non-commercial use. The Documentation may not be printed for resale or redistribution.
- b. For use of Documentation or Other Documentation where end user does not receive a EULA or does not have a valid license agreement with Intergraph, Intergraph grants the Licensee a non-exclusive license to use the Documentation or Other Documentation for Licensee's internal non-commercial use. Intergraph Corporation gives Licensee permission to print a reasonable number of copies of Other Documentation for Licensee's internal, non-commercial. The Other Documentation may not be printed for resale or redistribution. This license contained in this subsection b) may be terminated at any time and for any reason by Intergraph Corporation by giving written notice to Licensee.

Disclaimer of Warranties

Except for any express warranties as may be stated in the EULA or separate license or separate terms and conditions, Intergraph Corporation disclaims any and all express or implied warranties including, but not limited to the implied warranties of merchantability and fitness for a particular purpose and nothing stated in, or implied by, this document or its contents shall be considered or deemed a modification or amendment of such disclaimer. Intergraph believes the information in this publication is accurate as of its publication date.

The information and the software discussed in this document are subject to change without notice and are subject to applicable technical product descriptions. Intergraph Corporation is not responsible for any error that may appear in this document.

The software, Documentation and Other Documentation discussed in this document are furnished under a license and may be used or copied only in accordance with the terms of this license. THE USER OF THE SOFTWARE IS EXPECTED TO MAKE THE FINAL EVALUATION AS TO THE USEFULNESS OF THE SOFTWARE IN HIS OWN ENVIRONMENT.

Intergraph is not responsible for the accuracy of delivered data including, but not limited to, catalog, reference and symbol data. Users should verify for themselves that the data is accurate and suitable for their project work.

Limitation of Damages

IN NO EVENT WILL INTERGRAPH CORPORATION BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL INCIDENTAL, SPECIAL, OR PUNITIVE DAMAGES, INCLUDING BUT NOT LIMITED TO, LOSS OF USE OR PRODUCTION, LOSS OF REVENUE OR PROFIT, LOSS OF DATA, OR CLAIMS OF THIRD PARTIES, EVEN IF INTERGRAPH CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

UNDER NO CIRCUMSTANCES SHALL INTERGRAPH CORPORATION'S LIABILITY EXCEED THE AMOUNT THAT INTERGRAPH CORPORATION HAS BEEN PAID BY LICENSEE UNDER THIS AGREEMENT AT THE TIME THE CLAIM IS MADE. EXCEPT WHERE PROHIBITED BY APPLICABLE LAW, NO CLAIM, REGARDLESS OF FORM, ARISING OUT OF OR IN CONNECTION WITH THE SUBJECT MATTER OF THIS DOCUMENT MAY BE BROUGHT BY LICENSEE MORE THAN TWO (2) YEARS AFTER THE EVENT GIVING RISE TO THE CAUSE OF ACTION HAS OCCURRED.

IF UNDER THE LAW RULED APPLICABLE ANY PART OF THIS SECTION IS INVALID, THEN INTERGRAPH LIMITS ITS LIABILITY TO THE MAXIMUM EXTENT ALLOWED BY SAID LAW.

Export Controls

Intergraph Corporation's software products and any third-party Software Products obtained from Intergraph Corporation, its subsidiaries, or distributors (including any Documentation, Other Documentation or technical data related to these products) are subject to the export control laws and regulations of the United States. Diversion contrary to U.S. law is prohibited. These Software Products, and the direct product thereof, must not be exported or re-exported, directly or indirectly (including via remote access) under the following circumstances:

- a. To Cuba, Iran, North Korea, Sudan, or Syria, or any national of these countries.
- b. To any person or entity listed on any U.S. government denial list, including but not limited to, the U.S. Department of Commerce Denied Persons, Entities, and Unverified Lists, <http://www.bis.doc.gov/complianceand enforcement/liststocheck.htm>, the U.S. Department of Treasury Specially Designated Nationals List, www.treas.gov/offices/enforcement/ofac/http://www.pmddtc.state.gov/compliance/debar.html, and the U.S. Department of State Debarred List.
- c. To any entity when Licensee knows, or has reason to know, the end use of the Software Product is related to the design, development, production, or use of missiles, chemical, biological, or nuclear weapons, or other un-safeguarded or sensitive nuclear uses.
- d. To any entity when Licensee knows, or has reason to know, that an illegal reshipment will take place.

Any questions regarding export or re-export of these Software Products should be addressed to Intergraph Corporation's Export Compliance Department, Huntsville, Alabama 35894, USA.

Trademarks

Intergraph, the Intergraph logo, PDS, SmartPlant, FrameWorks, I-Sketch, SmartMarine, IntelliShip, ISOGEN, SmartSketch, SPOOLGEN, SupportManager, SupportModeler, and Intergraph Smart are trademarks or registered trademarks of Intergraph Corporation or its subsidiaries in the United States and other countries. Microsoft and Windows are registered trademarks of Microsoft Corporation. ACIS is a registered trademark of SPATIAL TECHNOLOGY, INC. Infragistics, Presentation Layer Framework, ActiveTreeView Ctrl, ProtoViewCtrl, ActiveThreed Ctrl, ActiveListBar Ctrl, ActiveSplitter, ActiveToolbars Ctrl, ActiveToolbars Plus Ctrl, and ProtoView are trademarks of Infragistics, Inc. Incorporates portions of 2D DCM, 3D DCM, and HLM by Siemens Product Lifecycle Management Software III (GB) Ltd. All rights reserved. Gigasoft is a registered trademark, and ProEssentials a trademark of Gigasoft, Inc. VideoSoft and VxFlexGrid are either registered trademarks or trademarks of ComponentOne LLC 1991-2014, All rights reserved. Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Tribon is a trademark of AVEVA Group plc. Alma and act/cut are trademarks of the Alma company. Other brands and product names are trademarks of their respective owners.

Contents

Basic Reporting.....	7
Adding a delivered report to the model	7
Adding report with default parameters	7
Editing Parameters.....	10
Creating Model Reports.....	12
Copy Report to Catalog	15
Adding Filter Based Query to Report	16
Adding Additional Property Queries	24
Allow Runtime Filter Selection.....	27
Grouping	29
Add Grouping to Report	29
Testing grouping	33
Using Delivered Macros	33
Sorting	34
Sorting- using queried property value	34
Sorting – using computed property value	35
Embedding output of a label in a filter based query	35
Formatting.....	37
Report Formatting using New or Existing Labels	37
New Format.....	37
Existing Label Format	40
Using a PHYSICAL token with UOM	42
Outputting codelisted values	52
Using a FORMAT_EXPRESSION.....	56
Conditional Formatting	59
Raw Values	59
Formatted Values	64
Creating a coordinate label.....	66
Creating the label	66
Editing Units of Measure	69
Editing the Coordinate System	70
COM Queries	71
Define a label query to report the parent system of equipment items.....	71
Combining Labels by editing RQE files	76
Recursive querying in labels.....	80
Recursive label using “Implements” ExitCondition	82
Multi-path label query	83
Multi-stroke relationship path in query and using edge	89
Label using multiple strokes.....	89
Label using Edge	92
Using Filter Condition in Query to Return Multiple Objects	95
Using Filter=“All”	95
Using Filter=“First” and “Last”	101
Creating a label that returns unit name for piping parts and piping welds	114

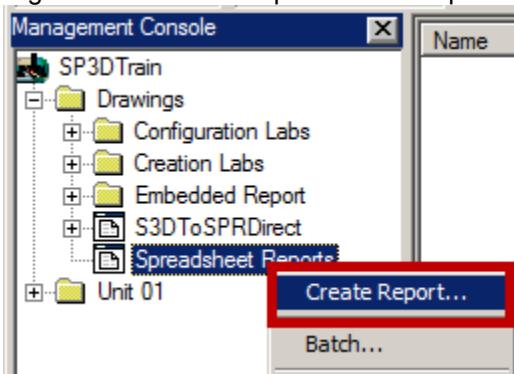
SQL Queries	115
Writing SQL Queries.....	115
Direct Property Queries.....	115
Codelisted attribute queries	117
Create SQL Label to return name of Equipment Parent	118
Optimizing the SQL Query and Label	122
Returning Substring of a property	122
Define SQL Filter.....	123
Creating SQL Reports	124
SQL Query Report	124
Filter Based SQL Query.....	128
Enable report to be embedded into a drawing.....	131
Test Report by Embedding	133
Advanced Reporting	136
Differential Reporting	136
Hierarchical Reporting	141
Delegated Query.....	145
Adding connected run name to nozzle report	145
Adding unit name label to piping MTO.....	146
Modifying Macros.....	147
Copy pipeline name to sheet name in multi-sheet report	147
Bulkloading reports and labels to catalog	151
Finding out internal name of manually created labels and reports.....	151
Loading additional label content from eCustomer.....	152
Embedding Labels	155
Pipeline Length Report using Run Length Label.....	155
Create Report.....	155
Edit Label	156
Edit the report.....	157
Embedding output of a label in SQL query.....	159
Embedding one label into another.....	160
S3DToSPRDirect Reports.....	163
Introduction	163
Creating S3DtoSPRDirect Report	164
Copy S3DToSPRDirect report to Catalog	167
Editing ReportSettings.txt	171
Adding Aspects	172
Adding Surface Style Rules	174

Basic Reporting

Adding a delivered report to the model

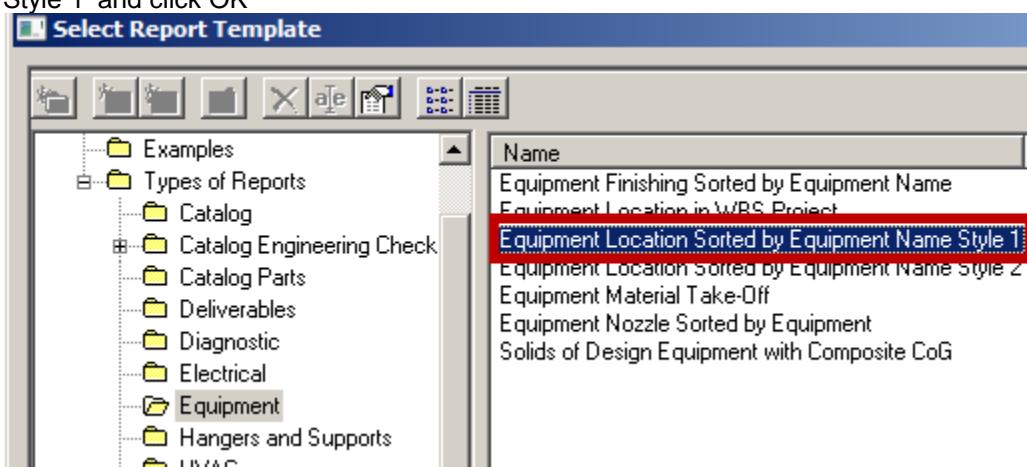
Adding report with default parameters

1. Start Smart 3D
2. Define workspace using Plant Filters - Training Filters – U01
3. Switch to the Drawings and Reports task
4. Right mouse click on Spreadsheet Reports and select 'Create Report'

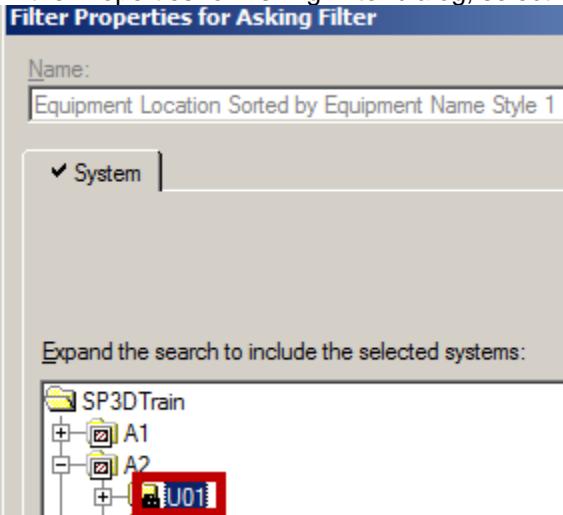


Note: Excel security settings must be set before a spreadsheet report can be created.

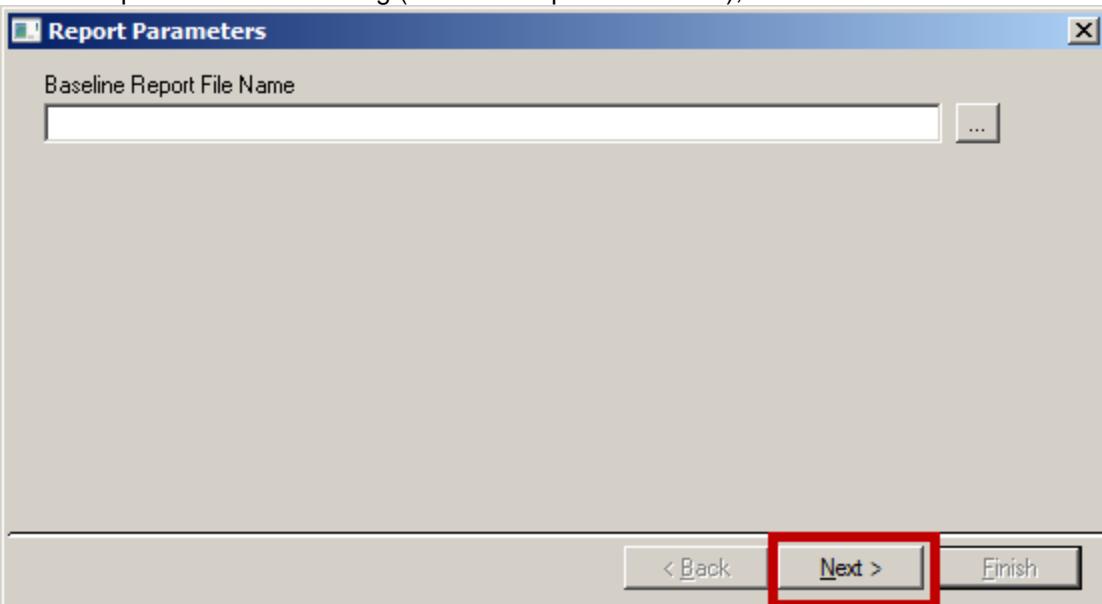
5. In the 'Select Report Template' dialog, select 'Equipment Location Sorted by Equipment Name Style 1' and click OK



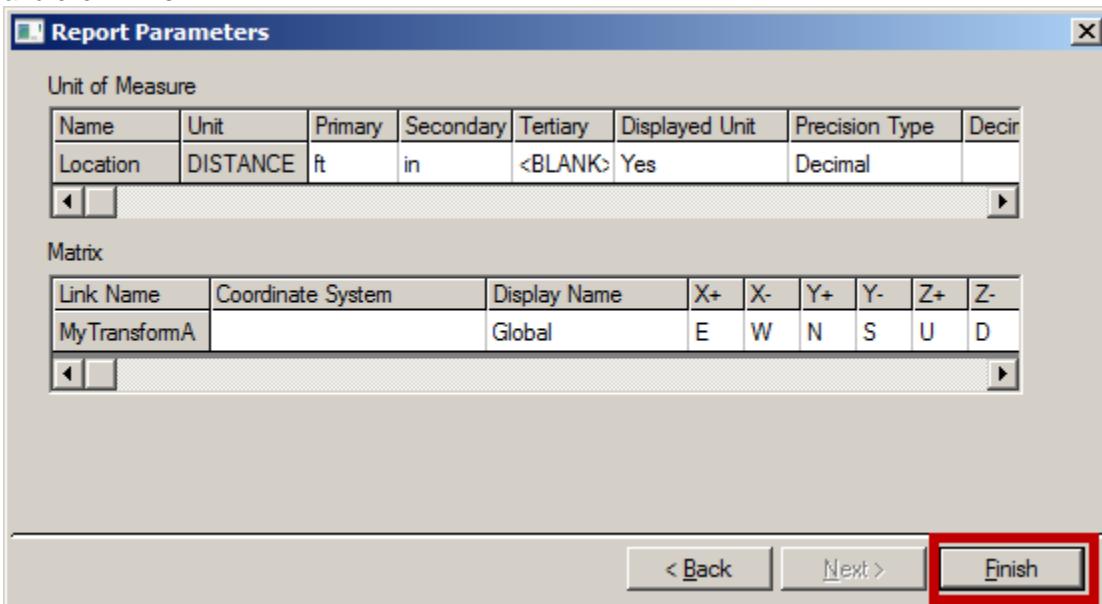
6. In the Properties for Asking Filter dialog, select U01 and click OK.



7. On the Report Parameters dialog (Baseline Report File Name), click 'Next'



8. On the next page of the report parameters dialog (UOM and Matrix), note the default parameters and click 'Finish'



9. Right click the report and Update Now
 10. Open the report after it is complete
 11. Notice the top left portion of the header shows the name of the report and the username of the person who updated it

C	D
Plant Name:	SP3DTrain
Filter Name:	Equipment Location Sorted by Equipment Nam
User:	INGRNET\User1

12. The top right portion of the header shows the date when the report was updated and the description of the filter (including the user selections)

G	H	I	J
Date:	10/23/2013 6:28:34 PM		
Filter Description:	[System IN (U01)] Object Type IN (Equipment&Furnishing\EquipmentTy		

13. The body of the report shows the coordinates in the Global coordinates in ft and in

Equipment	Location		
	ft in		
	Global		
E-102	E 85 ft 0 in	S 10 ft 0 in	U 6 ft 0 in
Electrical Device	E 85 ft 0 in	N 52 ft 0 in	U 3 ft 0 in

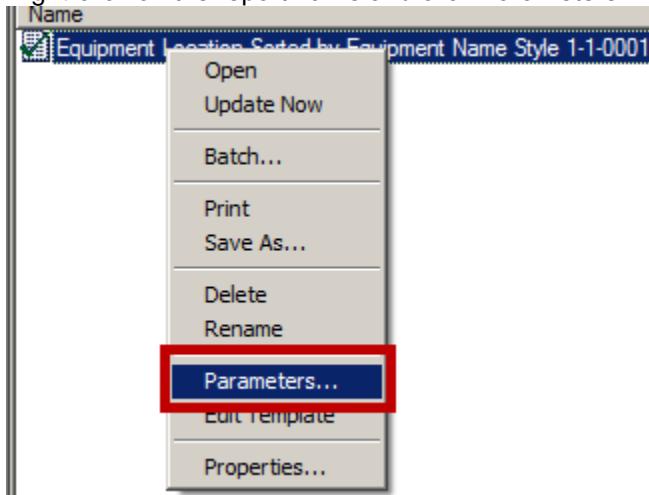
14. The right side of the report shows additional information and color coding based on approval status

Permission Group	Approval Status
Equipment	Working
Electrical	Approved
Equipment	Working
Equipment	Working

15. Close the excel report

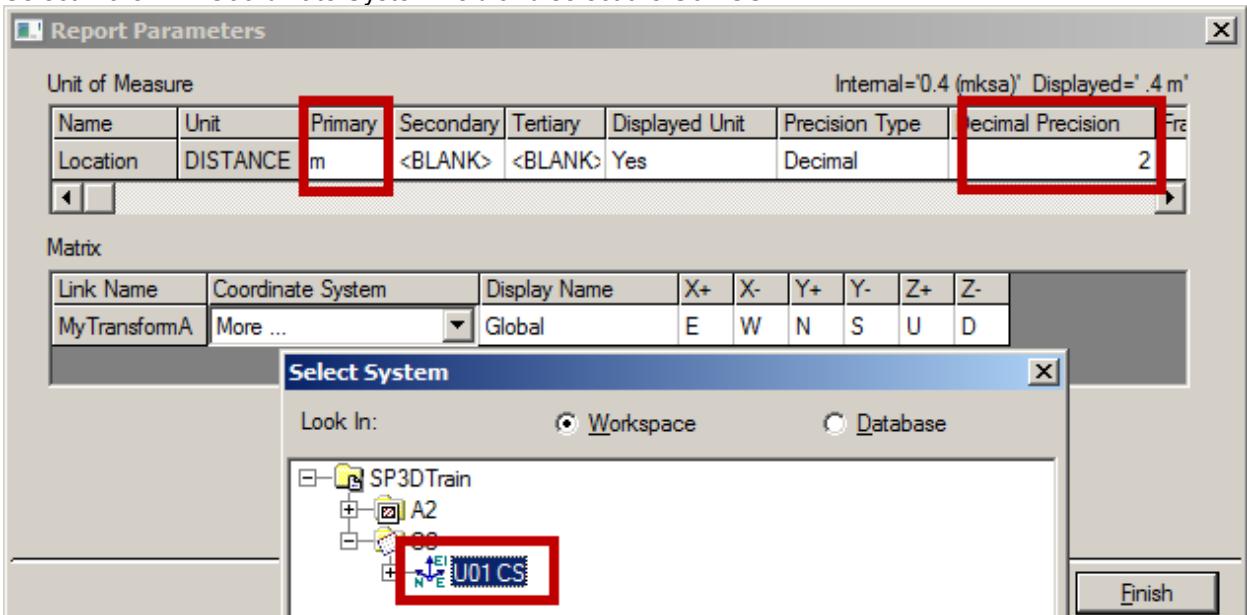
Editing Parameters

1. Right-click on the report name and click Parameters...

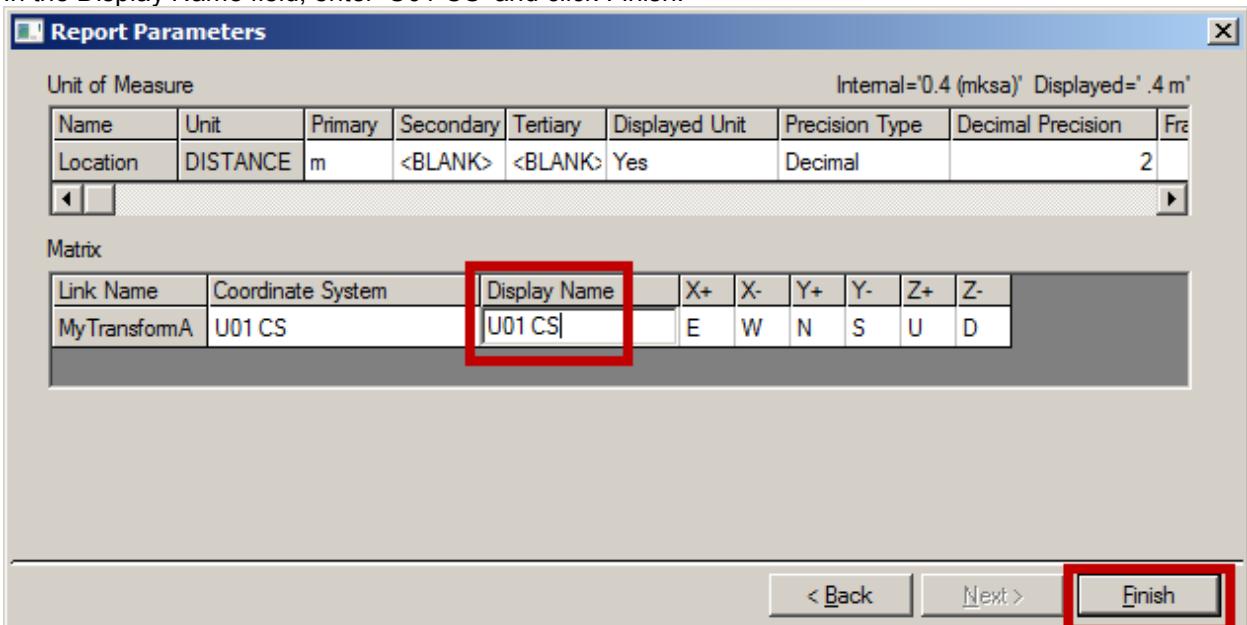


2. Select U01 and Next as before
3. In the last page of the report parameters dialog, change Primary to m and Decimal Precision to 2

4. Select More... in Coordinate System field and select the U01 CS



5. In the Display Name field, enter 'U01 CS' and click Finish.



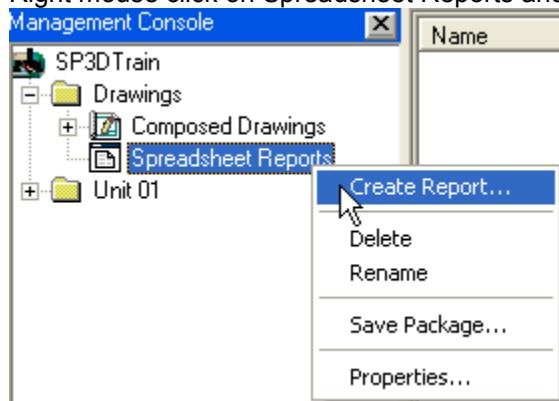
6. Update the report again and open it.
7. Notice that coordinates are now in m and w.r.t the U01 CS.

Location		
m		
U01 CS		
E 1.52 m	S 6.1 m	U 1.83 m
E 1.52 m	N 12.8 m	U .91 m

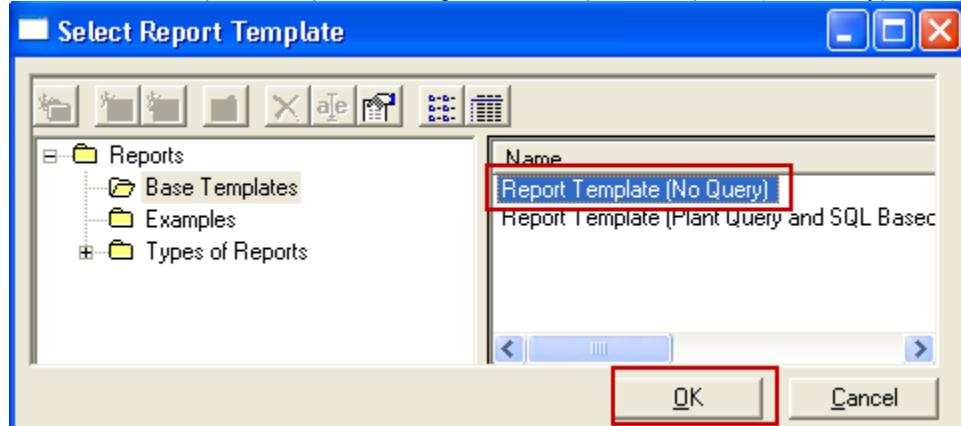
8. Close Excel

Creating Model Reports

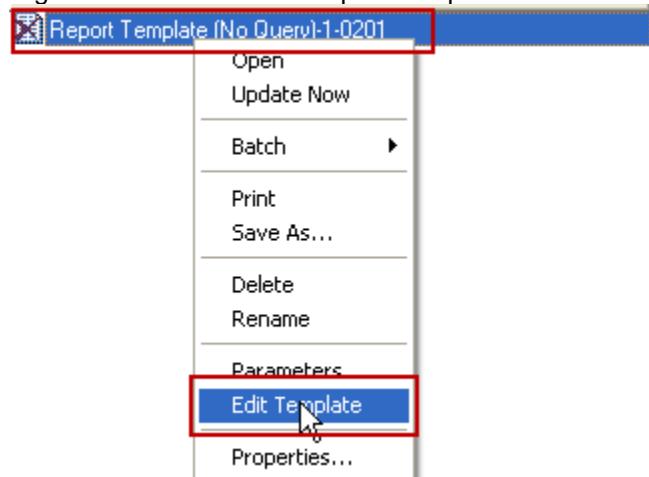
1. Right mouse click on Spreadsheet Reports and select 'Create Report'



2. In the 'Select Report Template' dialog, select 'Report Template (No Query)' and click OK.



3. Right mouse click on the Report Template in the Detail view and select 'Edit Template'



Note: To edit a report spreadsheet, it is required to be in the Drawings and Reports task.

4. Select the 'Design Layout' button on the Formatting tab.



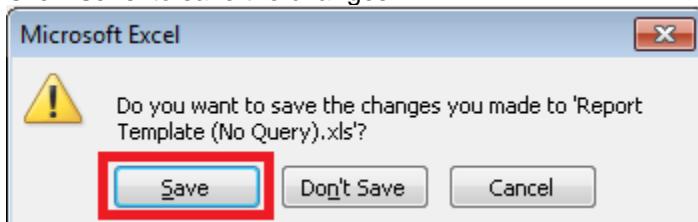
5. Excel opens and the Report Template (No Query).xls and Design Layout browser are shown.

6. Select 'Date' from the design layout browser and drag it to cell B1. Cell B1 now shows the field codes that are interpreted during report execution to display the date.

7. Drag items and enter text in column A till your display is as shown below

8. Close excel (using the close box, File → Exit or equivalent)

9. Click 'Save' to save the changes.

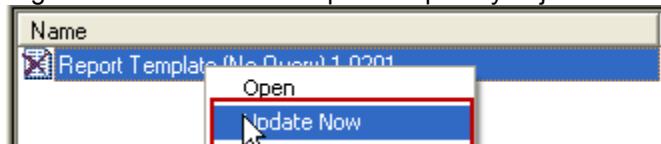


10. Select File → Save Report Template to save the report template into the model database.

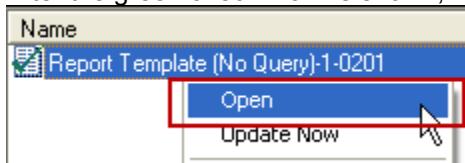
11. Close the report template using the lower close box.



12. Right mouse click on the report template you just modified and select 'Update Now'



13. After the green check mark is shown, right mouse click and select 'Open'



14. Report will look similar to below

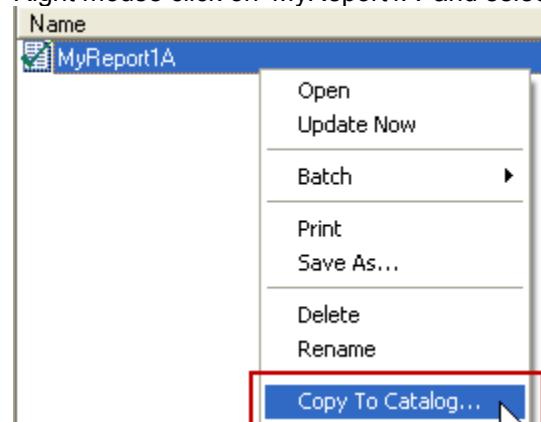
Report Template (No Query)-1-0201.xls [Read-Only] [Con]		
	B	C
1	Date	9/3/2009 4:24:12 PM
2	User Name	INGRNET\sdongre
3	Report Name	Report Template (No Query)
4		
5		

15. Close Excel.

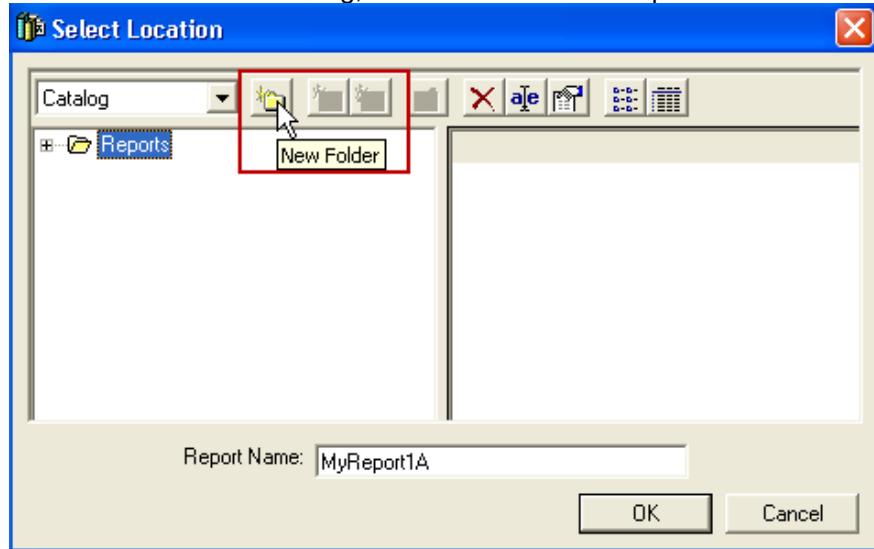
16. Rename the report to 'MyReport1A'

Copy Report to Catalog

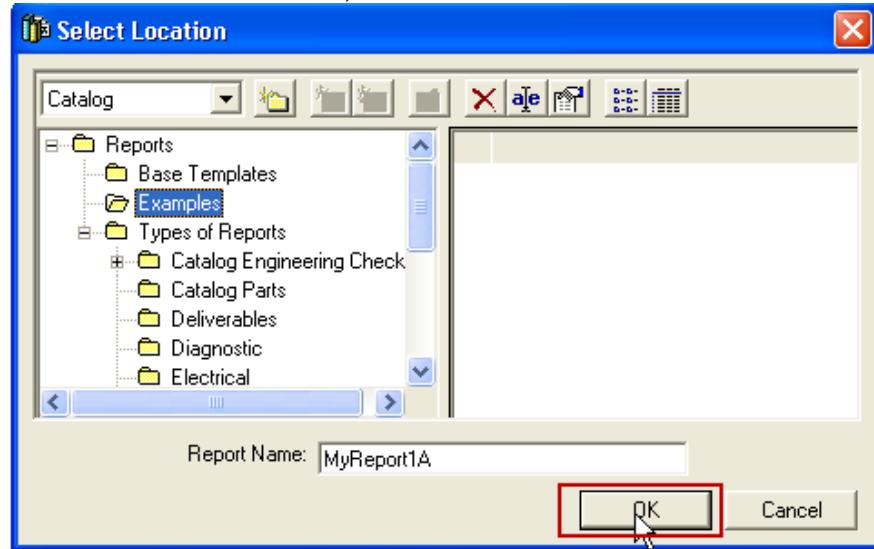
1. Right mouse click on 'MyReport1A' and select 'Copy to Catalog'



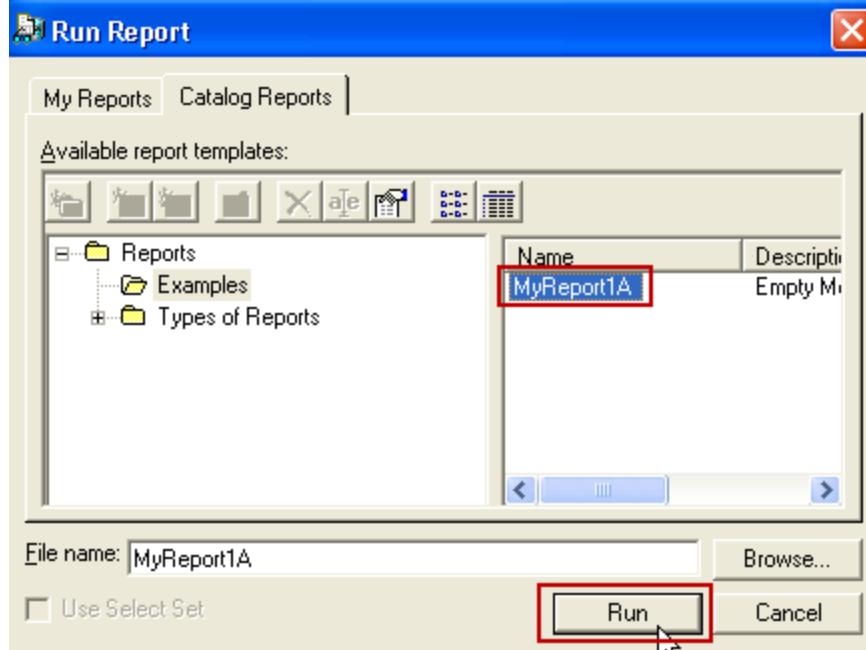
2. In the 'Select Location' dialog, select the root node 'Reports' and click the 'New Folder' button



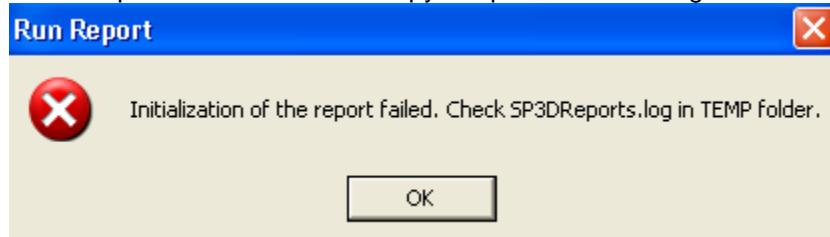
3. A new folder is created. Name the folder 'Examples'
4. With the new folder selected, click 'OK'



5. Select Tasks → Common to switch to the Common task.
6. Select Tools → Run Report
7. In the 'Run Report' dialog, select the 'MyReport1A' report and click Run.



8. Since the report does not contain a query, an error message is shown. This is expected behavior. This completes the workflow to copy a report to the catalog.

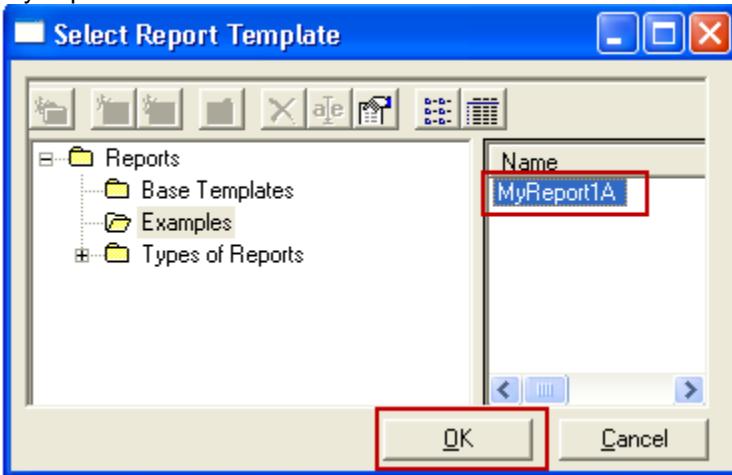


9. Click OK to dismiss the message and Cancel to dismiss the run report dialog.
10. The Copy to Catalog command has created an entry in the catalog reports hierarchy and copied files over to the symbols share. To verify, browse to your symbols share and you should see the hierarchy SharedContent\Reports\Examples\MyReport1A

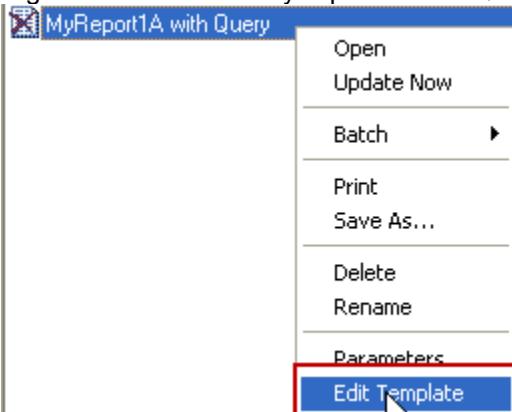
Adding Filter Based Query to Report

1. Switch to Drawings and Reports task
2. Right mouse click on 'Spreadsheet Reports' and select 'Create Report'

3. Expand the Reports\Examples tree and select 'MyReport1A' and click OK. This will create MyReport1A-0001 in the detail view.

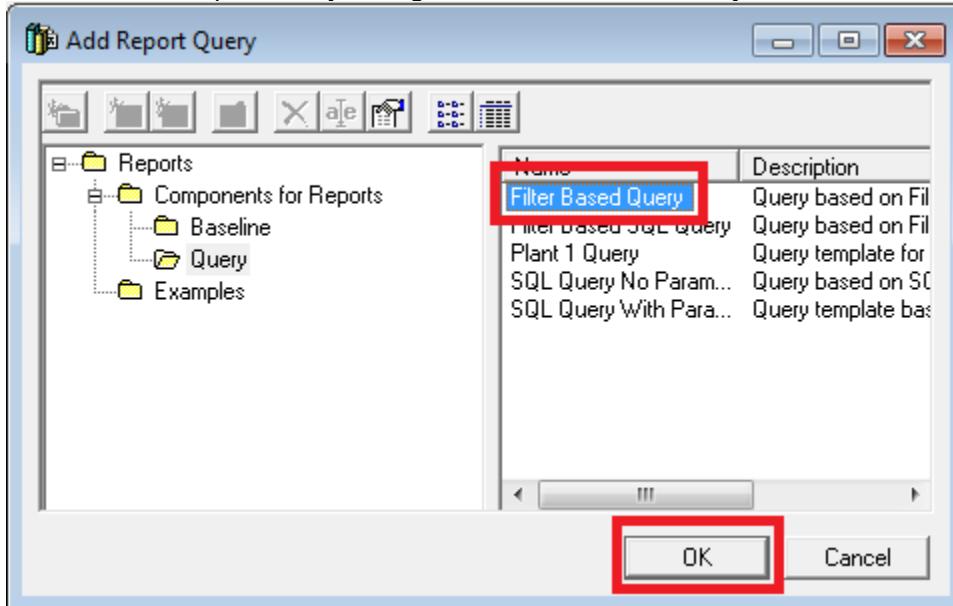


4. Rename 'MyReport1A-0001' to 'MyReport1A with Query'
5. Right mouse click on 'MyReport1A with Query' and Edit Template

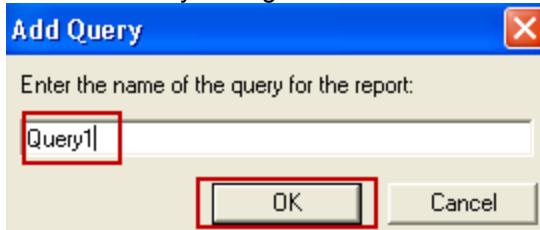


6. Select Tools → Add Query

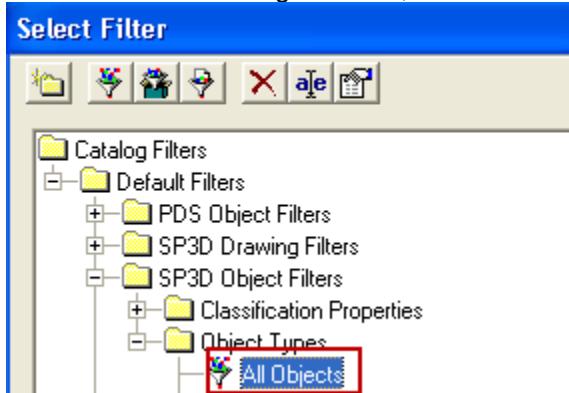
7. From the 'Add Report Query' dialog, select 'Filter Based Query' and click OK.



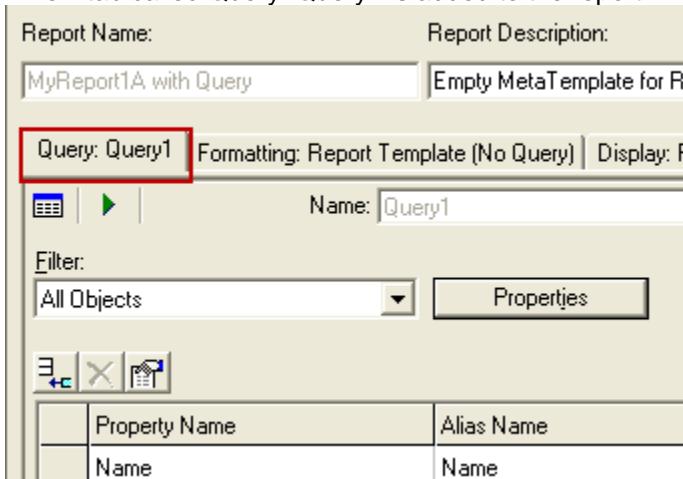
8. The 'Add Query' dialog is shown. Enter 'Query1' and click 'OK'



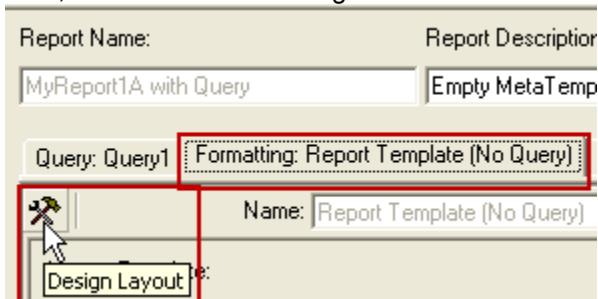
9. The 'Select Filter' dialog is shown, select the 'All Objects' filter and click OK.



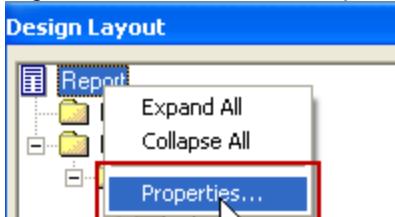
10. A new tab called Query: Query1 is added to the report.



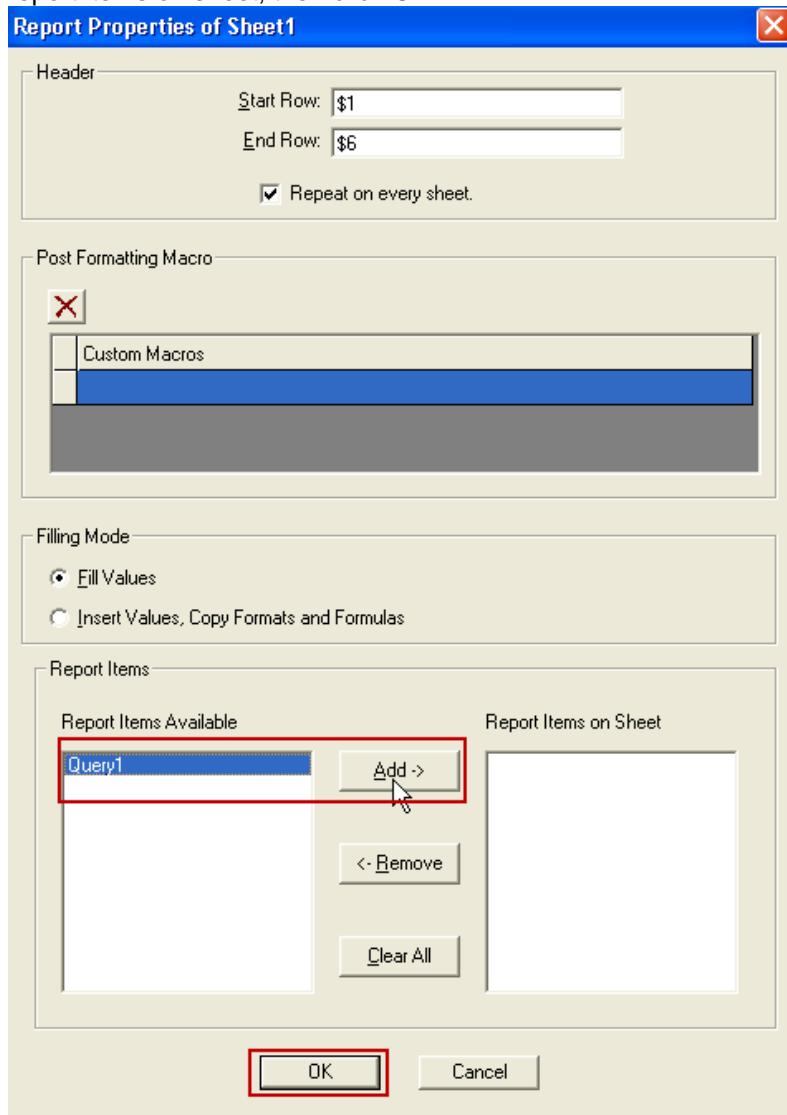
11. Now, click on the 'Formatting' tab and click the 'Design Layout' button. This opens Excel.



12. Right mouse click on the 'Report' node in the Design Layout browser and click 'Properties...'



13. This opens the report properties of Sheet1 dialog, select the query Query1 and Add it to the report items on sheet, then click OK.



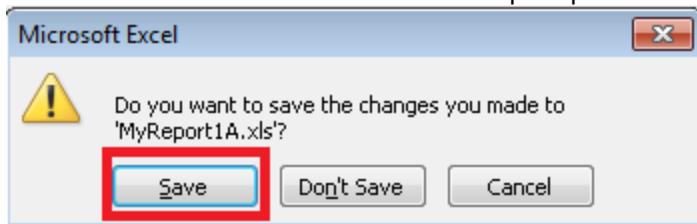
14. The Query1 node now appears in the tree. Drag 'FilterName' to cell E1 and 'FilterDescription' to cell E2. Then type in the text in cells D1 and D2

D	E
Filter Name	#Query1:{FilterName}#
Filter Description	#Query1:{FilterDescription}#

15. Type 'Name' in cell D7 and 'OID' in cell E7. Then drag and drop Name to cell D9 and OID to cell E9

D	E
1 Filter Name	#Query1::{FilterName}#
2 Filter Description	#Query1::{FilterDescription}#
3	
4	
5	
6	
7 Name	
8	
9 #Query1::Name#	#Query1::OID#

16. Close Excel and click 'Save' to save when prompted.



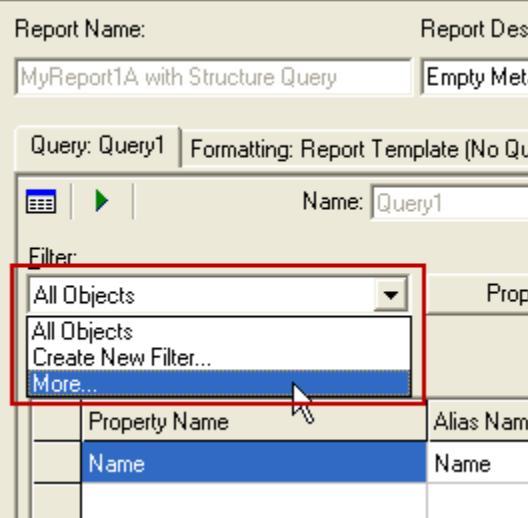
17. Save the report template using File → Save Report Template
18. Close the report template using the lower close box.
19. Right mouse click on 'MyReport1A with Query' and update it.

20. Open the report using the Open command or by double-clicking the report.

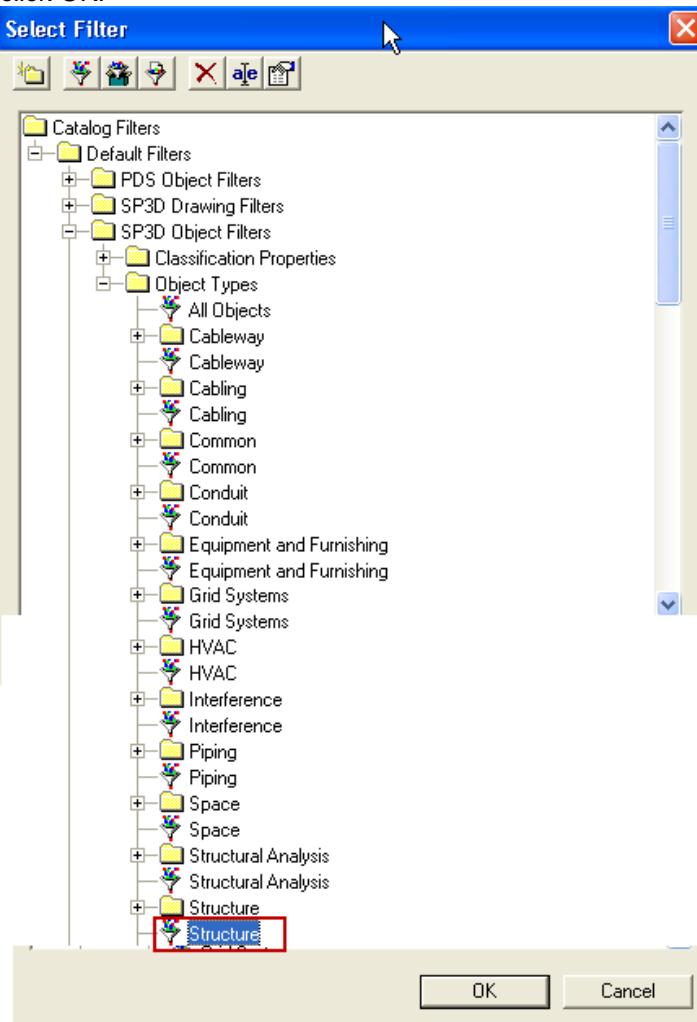
D	E	F
1	Filter Name	All Objects
2	Filter Description	Object Type IN (GridSystems) AND (HgrSupports
3		
4		
5		
7	Name	OID
9	Operator Occurrence	{0000272B-0000-0000-1A0A-2BE7C8492604}
10	Operator Occurrence	{0000272B-0000-0000-3D34-333B7E493004}
11	Operator Occurrence	{0000272B-0000-0000-2C20-33E47D495F04}
12	Operator Occurrence	{0000272B-0000-0000-4619-33E47D495F04}
13	Operator Occurrence	{0000272B-0000-0000-5D1F-33E47D495F04}
14	Operator Occurrence	{0000272B-0000-0000-9517-33E47D495F04}
15	Operator Occurrence	{0000272B-0000-0000-AC26-33E47D495F04}
16	Operator Occurrence	{0000272B-0000-0000-C221-33E47D495F04}
17	Operator Occurrence	{0000272B-0000-0000-8508-85ACBA492404}
18	Operator Occurrence	{0000272B-0000-0000-DC03-8C76BA492404}
19	Operator Occurrence	{0000272B-0000-0000-0B01-90B5EB4E9C19}
20	Operator Occurrence	{0000272B-0000-0000-0C01-90B5EB4E9C19}
21	Operator Occurrence	{0000272B-0000-0000-0D01-90B5EB4E9C19}
22	Operator Occurrence	{0000272B-0000-0000-0E01-90B5EB4E9C19}
23	Operator Occurrence	{0000272B-0000-0000-1C01-90B5EB4E9C19}
24	Operator Occurrence	{0000272B-0000-0000-3C1A-A16CC1492604}
25	Operator Occurrence	{0000272B-0000-0000-5C0D-B225D2492804}

21. Close Excel.
22. Right mouse click on 'MyReport1A with Query' and 'Copy to Catalog'
23. Select the 'Examples' folder and click 'OK'
24. Switch to Common task using Tasks → Common
25. Select Tools → Run Report
26. Select the 'MyReport1A with Query' and click Run. The report will run and open on your screen.
27. Review the report and close Excel.
28. Switch to Drawings and Report task.
29. Right mouse click on 'Spreadsheet Reports' and select 'Create Report'
30. Expand the Reports\Examples tree and select 'MyReport1A with Query' and click OK. This creates a new report in the detail view.
31. Rename the report to 'MyReport1A with Structure Query' and Edit Template.

32. Select the Filter drop down and select More...



33. Select the 'Catalog Filters\Default Filters\SP3D Object Filters\Object Types\Structure' filter and click OK.

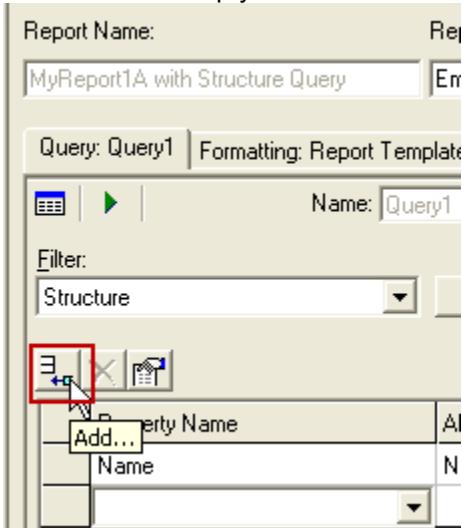


34. File → Save Report Template and close the report template.

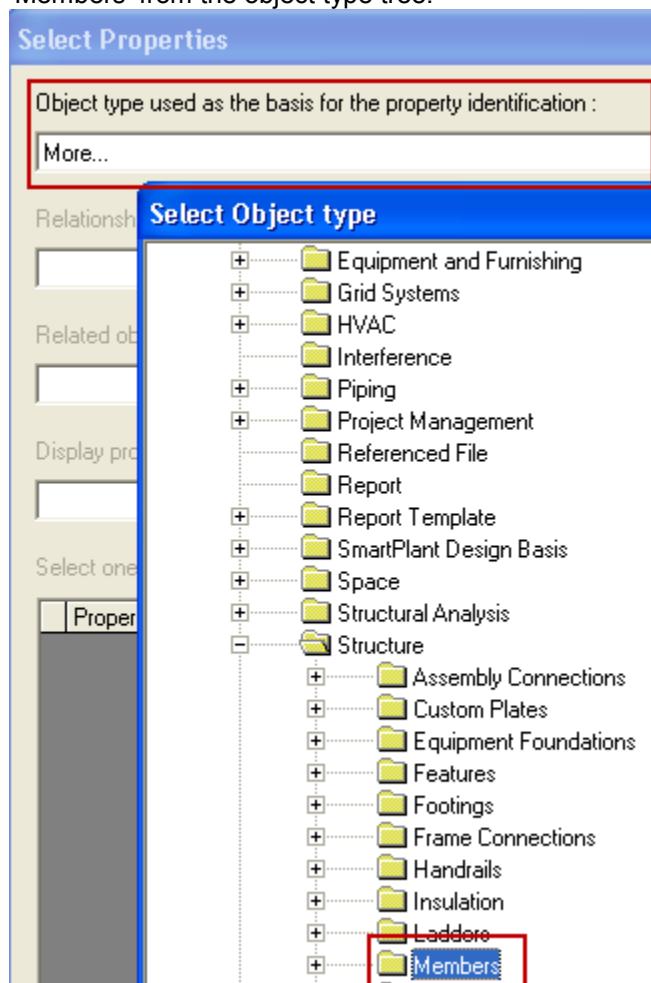
35. Update the report using 'Update Now' and then view it. Notice only structural objects are shown.

Adding Additional Property Queries

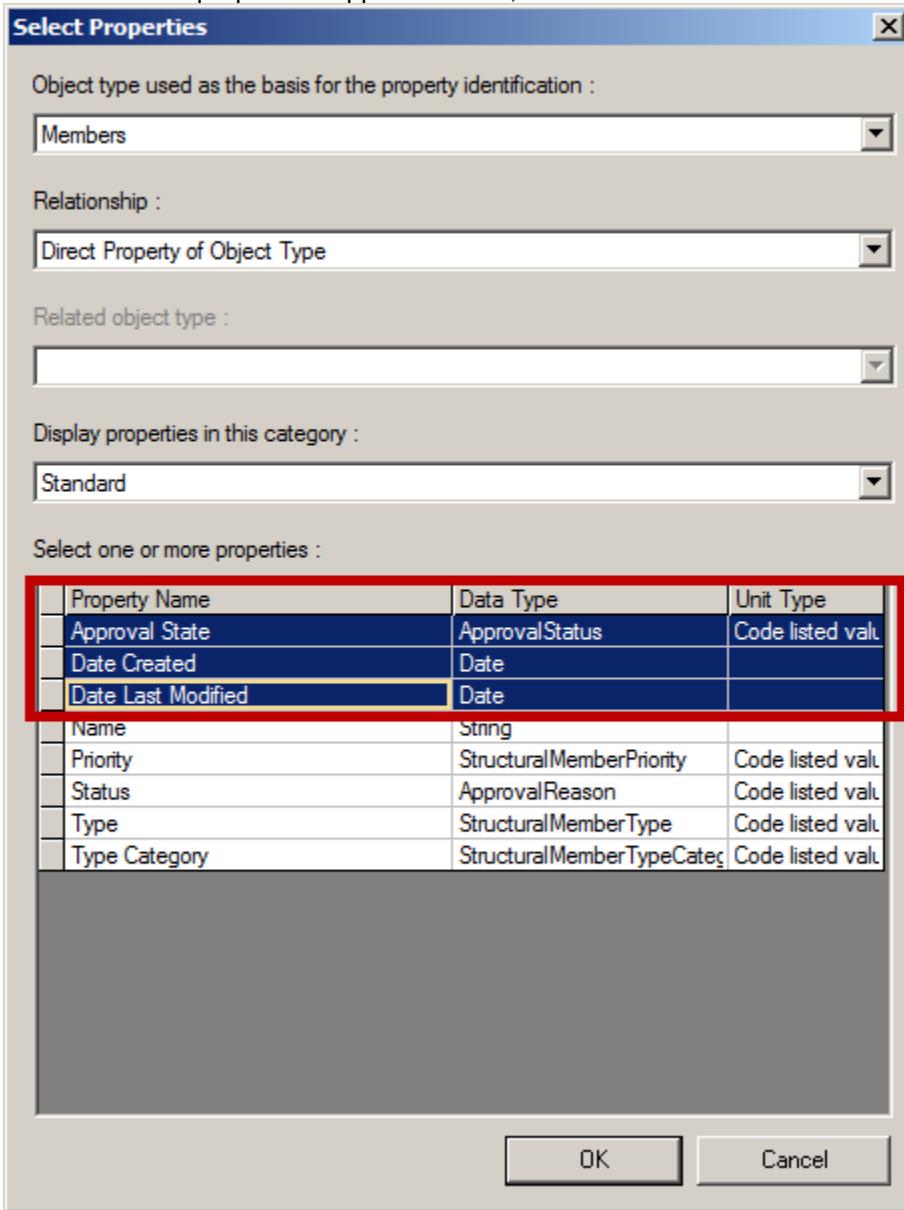
1. Right mouse click on the 'My Report1A with Structure Query' and Edit Template
2. Select the first empty row and click the 'Add...' button



3. Select the topmost drop down (object type used as the basis for property identification) and select 'Members' from the object type tree.



4. Select the three properties 'Approval State', 'Date Created' and 'Date Last Modified' and click OK.



5. Select the 'Formatting' tab and click 'Design Layout'

6. Drag the properties to the cells F9, G9 and H9 and add headers in F7, G7 and H7 as shown

F	G	H	
1			
2			
3			
4			
5			
6			
7	Created	Modified	Status
8	#Query1::Date Created#Query1::Date Last Modified#Query1::Approval State#		
9			
10			

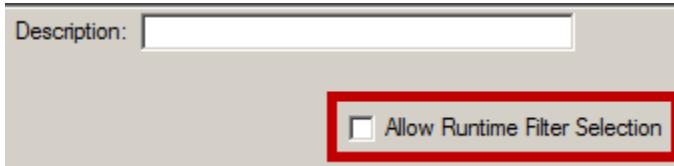
7. Close Excel and save when prompted.

8. File → Save Report Template to save the report template and close the template using close box.

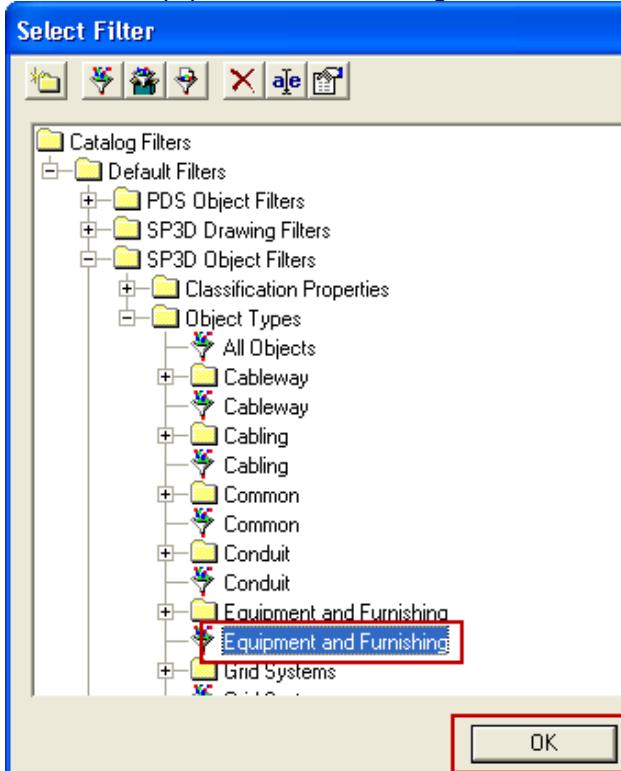
9. Update the report using 'Update Now' command.
10. Right mouse click on 'MyReport1A with Structure Query' and select 'Copy to Catalog'
11. Select the Reports\Examples folder and click OK to copy report to catalog.

Allow Runtime Filter Selection

1. Right mouse click on 'Spreadsheet Reports' and select 'Create Report'
2. Expand the Reports\Examples tree and select 'MyReport1A with Structure Query' and click OK.
3. Rename the newly created report to 'MyReport1A with Runtime Filter' and click OK
4. Right click on the newly created report and Edit Template
5. Check the 'Allow Runtime Filter Selection' box



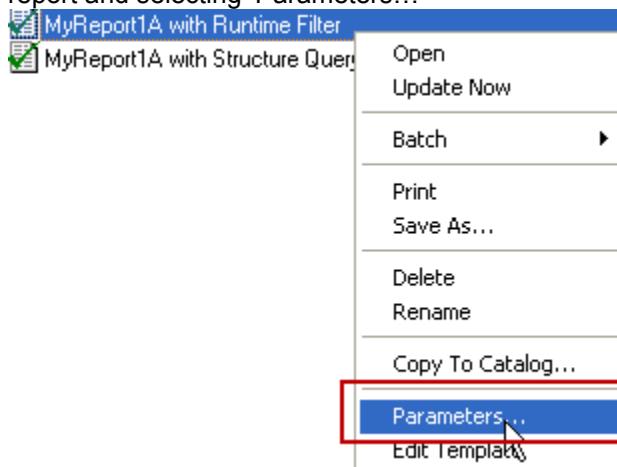
6. File → Save Report Template , you are prompted to select a filter.
7. Select the 'Equipment and Furnishings' filter and click OK.



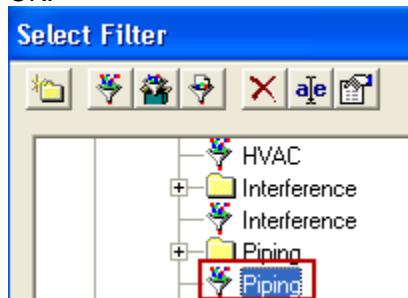
8. Close the report template using the close box.
9. Update the report using update now command and view the report.
10. Notice that the filter name you picked is shown in the header of the report and all the objects are equipment objects such as nozzles, shapes, equipment components and equipment.

	D	E	F
1	Filter Name	Equipment and Furnishing	
2	Filter Description	Object Type IN (Equipment&Furnishing)	
3			
4			
5			
6			
7	Name	OID	Created
8			
9	N1	(00004E23-0000-0000-2B001/22/2009 7:15:28	
10	F	(00004E23-0000-0000-3B001/22/2009 7:56:48	
11	N2	(00004E23-0000-0000-3F001/22/2009 7:17:38	
12	N1	(00004E23-0000-0000-65001/22/2009 7:29:38	
13	Suction	(00004E23-0000-0000-8D003/24/2009 9:41:48	
14	Discharge	(00004E23-0000-0000-95003/24/2009 9:41:48	

11. Right mouse click on 'MyReport1A with Runtime Filter' and select 'Copy to Catalog'
12. Select the Reports\Examples folder and click OK to copy report to catalog.
13. The filter is now a parameter of the report. It may be changed at any time by right clicking the report and selecting 'Parameters...'



14. Select the 'Catalog Filters\Default Filters\SP3D Object Filters\Object Types\Piping' filter and click OK.

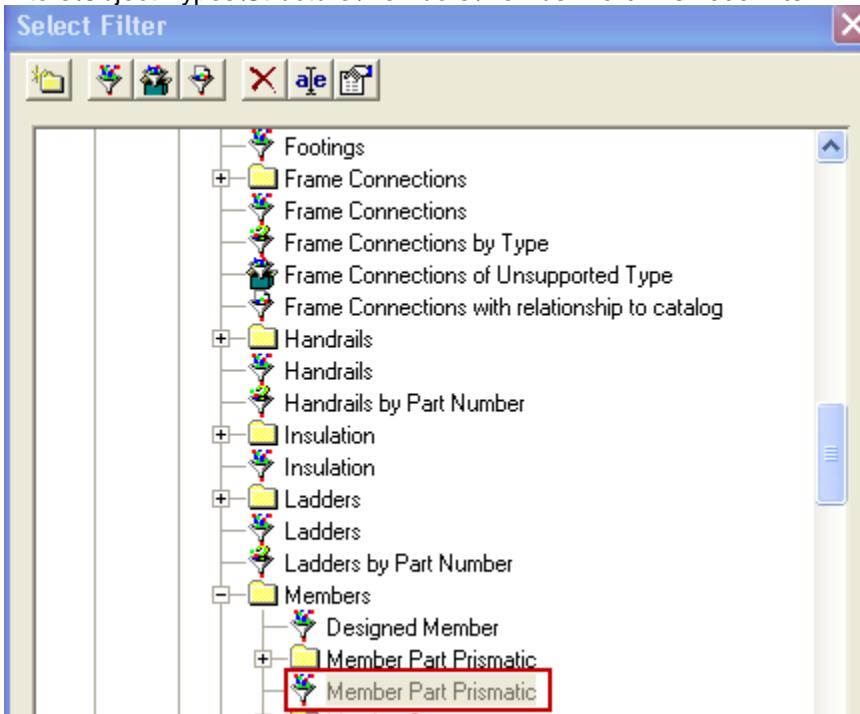


15. Update the report and observe the results.

Grouping

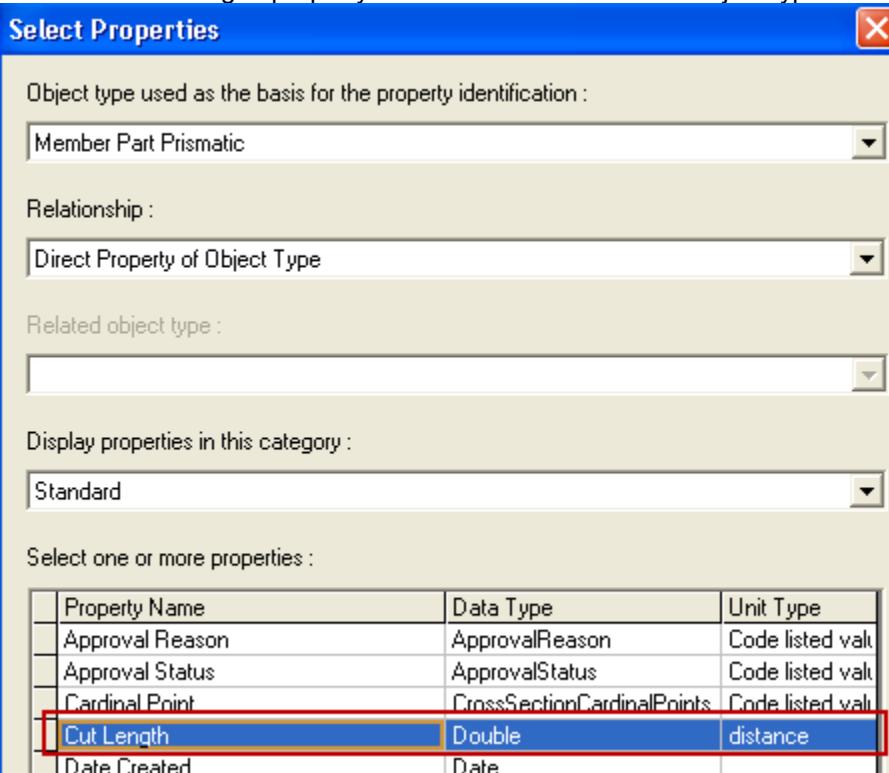
Add Grouping to Report

1. Right mouse click on 'Spreadsheet Reports' and select 'Create Report'
2. Expand the Reports\Examples tree and select 'MyReport1A with Structure Query' and click OK. This will create a new report in the detail view.
3. Rename the report to 'MyReport1A with Grouping'
4. Right mouse click on 'MyReport1A with Grouping' and Edit Template.
5. In the 'Filter' field, select More... and then select the 'Catalog Filters\Default Filters\SP3D Object Filters\Object Types\Structure\Members\Member Part Prismatic' filter

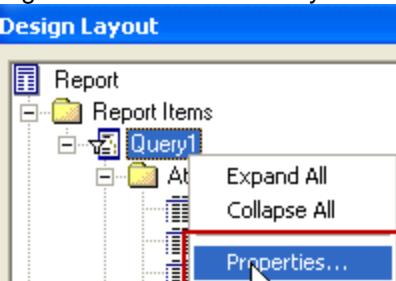


6. Click 'Add...' to add a new property to the query

7. Select the 'Cut Length' property for Member Part Prismatic object type.

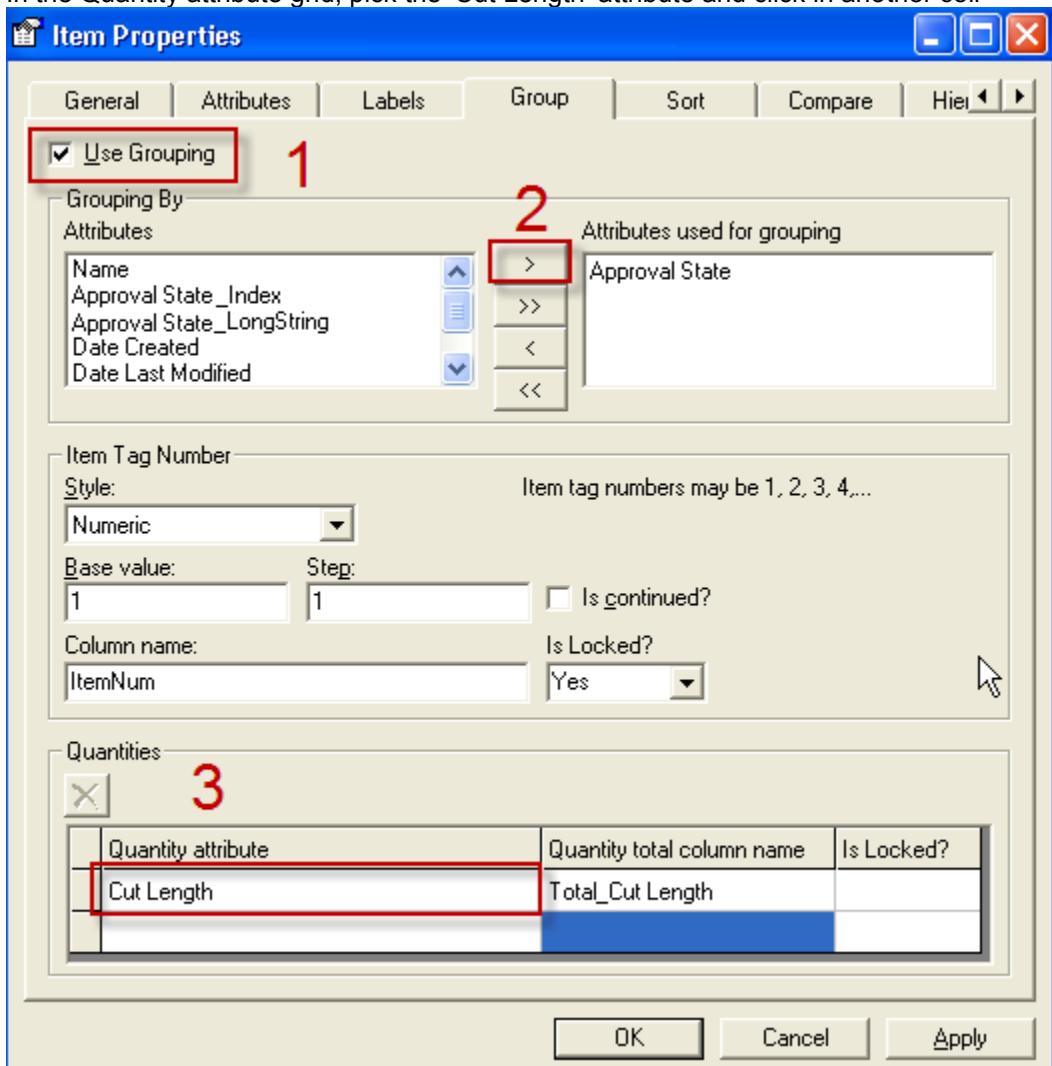


8. Click the Formatting tab and enter Design Layout.
9. Right mouse click on Query1 and select Properties



10. Click the Group tab
11. Check the 'Use Grouping' box
12. In the Grouping By Attributes list, select Approval State and click the > sign to add it to the 'Attributes used for grouping'

13. In the Quantity attribute grid, pick the 'Cut Length' attribute and click in another cell



14. Click OK – you will notice a new branch 'Grouping' is added to the tree view

15. Drag and drop 'Number – ItemNum' to cell C9

The screenshot shows the 'Design Layout' window with the report structure on the left and an Excel spreadsheet on the right. The report items tree includes 'Query1' which contains 'Attributes' (Name, Approval Status, Approval Status_Index, Approval Status_LongString, Date Created, Date Last Modified, Cut Length, OID) and 'Grouping' (Quantity - Total_Cut Length, Number - ItemNum). The Excel spreadsheet has columns C, D, E, F, G, H, I, J, K, L and rows 1 through 17. Cell C9 contains the formula '#Query1::ItemNum#'. A red arrow points from the 'Number - ItemNum' item in the report items tree to cell C9.

16. Drag and drop 'Cut Length' to cell I9 and 'Quantity – Total_Cut Length' to cell J9

The screenshot shows the 'Design Layout' window with the report structure on the left and an Excel spreadsheet on the right. The report items tree includes 'Query1' which contains 'Attributes' (Name, Approval Status, Approval Status_Index, Approval Status_LongString, Date Created, Date Last Modified, Cut Length, OID) and 'Grouping' (Quantity - Total_Cut Length, Number - ItemNum). The Excel spreadsheet has columns I, J, K, L and rows 1 through 17. Cells I9 and J9 contain the formulas '#Query1::Cut Len#' and '#Query1::Total_Cut Length#' respectively. A red arrow points from the 'Cut Length' item in the report items tree to cell I9.

17. Use Excel functions such as Format Painter to edit the headers to suit.

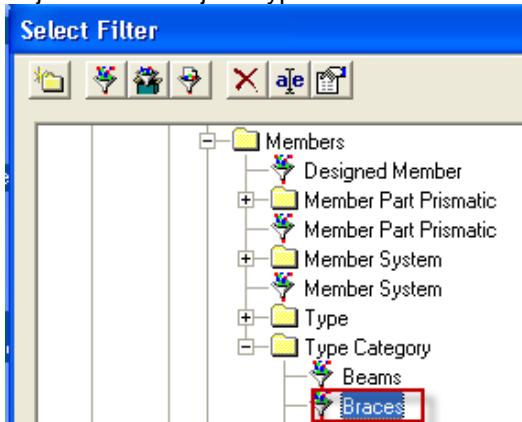
The screenshot shows the Excel spreadsheet with the header row highlighted. The first two columns are outlined with a red border and have a yellow background. They are labeled 'Cut Length' and 'Total Cut Length' respectively. The formula '#Query1::Cut Len#' is in cell I9 and '#Query1::Total_Cut Length#' is in cell J9.

18. Close Excel and save when prompted.

19. File → Save Report Template
20. Close the report template using the lower close box.
21. Update report and observe results – in the training plant, all members are in working status.

Testing grouping

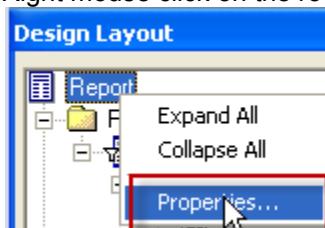
22. Switch to the Structure task and define workspace using filter Plant Filters\Training Filters\U02
23. Start Tools → Select by Filter command and select the filter 'Catalog Filters\Default Filters\SP3D Object Filters\Object Types\Structure\Members\Type Category\Braces' and click OK.



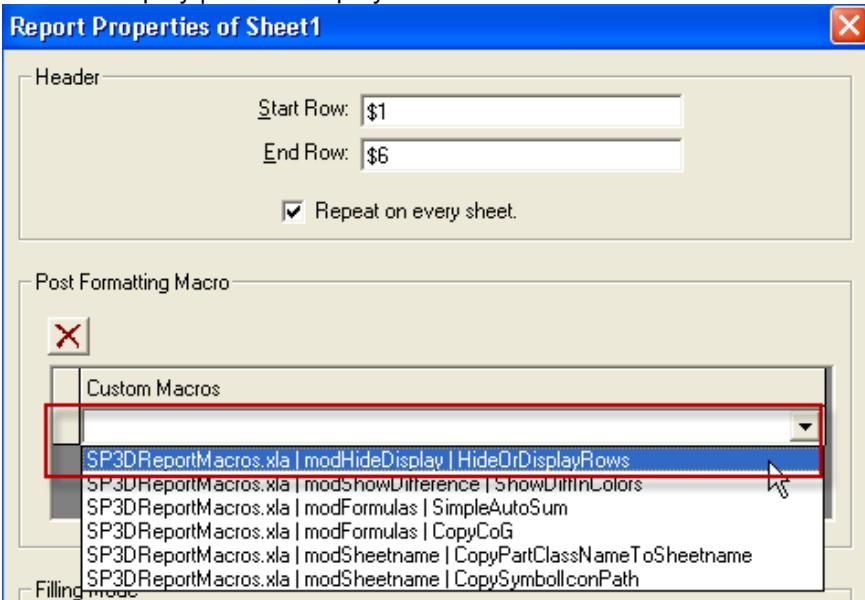
24. Edit → Properties and switch to Configuration tab.
25. Select status 'Rejected' and click OK.
26. Similarly change the status of all Beams to 'Approved'.
27. Switch to the Drawings and Reports task.
28. Right mouse click on 'MyReport1A with Grouping' and 'Update Now'

Using Delivered Macros

29. Right mouse click on 'MyReport1A with Grouping' and 'Edit Template'
30. Click the Formatting tab and enter design layout.
31. Right mouse click on the root node 'Report' and select Properties



32. Click in the first empty row of the ‘Custom Macros’ grid and select ‘SP3DReportMacros.xls | modHideDisplay | HideOrDisplayRows’

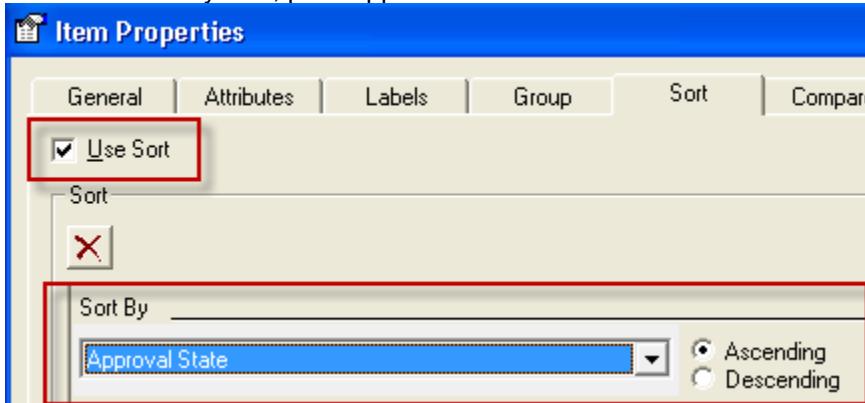


33. Click OK

Sorting

Sorting- using queried property value

1. Right mouse click on ‘Query1’ and click Properties
2. Click the ‘Sort’ tab
3. Check the ‘Use Sort’ check box
4. In the first Sort By field, pick ‘Approval State’



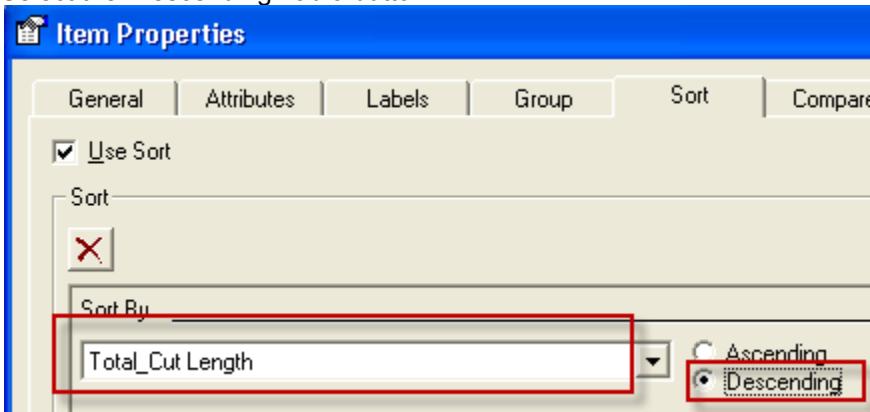
5. Click OK
6. Close Excel and save when prompted.
7. File → Save Report Template
8. Close the report template using the lower close box.

9. Update report and observe results – numbers should be similar to below

Status	Cut Length	Total Cut Length
Approved	4.538599	187.887102
Rejected	4.461419173	37.54306987
Working	3.88874	1463.547019

Sorting – using computed property value

10. Right mouse click on 'MyReport1A with Grouping' and 'Edit Template'
11. Click the Formatting tab and enter design layout.
12. Right mouse click on 'Query1' and click Properties
13. Click the 'Sort' tab
14. In the 'Sort By' field, select Total_Cut Length
15. Select the 'Descending' radio button



16. Click OK.
17. Close Excel and save when prompted.
18. File → Save Report Template
19. Close the report template using the lower close box.
20. Update report and observe results – now the order is different with the working status members in the first row.

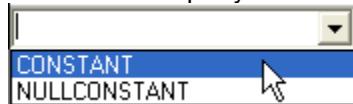
Status	Cut Length	Total Cut Length
Working	3.88874	1463.547019
Approved	4.538599	187.887102
Rejected	4.461419173	37.54306987

21. Right mouse click on 'MyReport1A with Grouping' and select 'Copy to Catalog'
22. Select the Reports\Examples folder and click OK to copy report to catalog.

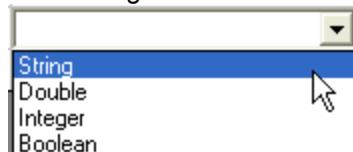
Embedding output of a label in a filter based query

1. Right mouse click on 'Spreadsheet Reports' and select 'Create Report'
2. Expand the Reports\Examples tree and select 'MyReport1A with Grouping' and click OK. This will create a new report in the detail view.
3. Rename the report to 'MyReport1A with Embedded Label Query'
4. Right mouse click on 'MyReport1A with Embedded Label Query' and Edit Template.

- Click in the Property Name column in the last empty row in the grid and select CONSTANT



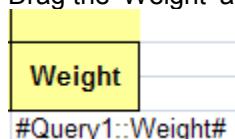
- Enter 'Weight' in the 'Alias Name' cell
- Pick 'String' as the Constant Type



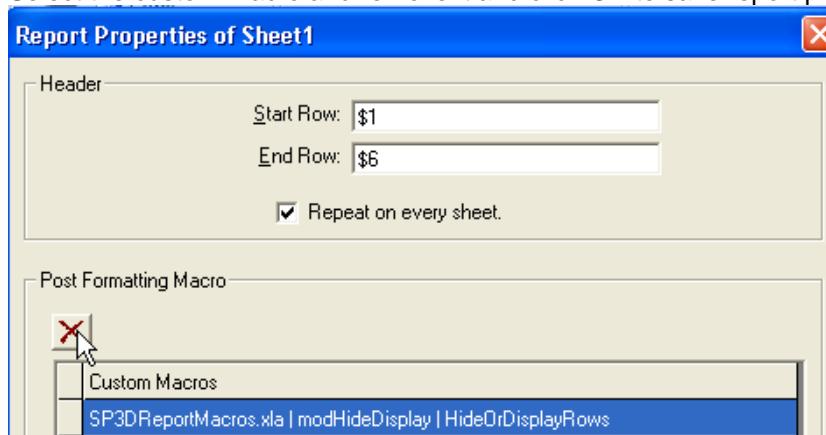
- Enter [Structure Member Weight] (including the rectangular brackets) in the Constant Value column

CONSTANT	Weight	String	[Structure Member Weight]
----------	--------	--------	---------------------------

- Click the formatting tab and enter design layout.
- Drag the 'Weight' attribute to cell L9 and adjust the header to suit.



- Right mouse click on the Report and select properties.
- Select the custom macro and remove it and click OK to save report properties.



- Close Excel and save when prompted.
- File → Save Report Template
- Close the report template using the lower close box.
- Update report and observe results – should be similar to below

Status	Cut Length	Total Cut Length	Weight
Working	3.88874	1463.547019	428.63 lbm
Working	7.845565518	1463.547019	575.14 lbm
Working	7.845565518	1463.547019	575.14 lbm
Working	7.845879273	1463.547019	575.16 lbm
Working	0.82677	1463.547019	24.22 lbm

- Right mouse click on 'MyReport1A with Embedded Label Query' and select 'Copy to Catalog'
- Select the Reports\Examples folder and click OK to copy report to catalog.

Formatting

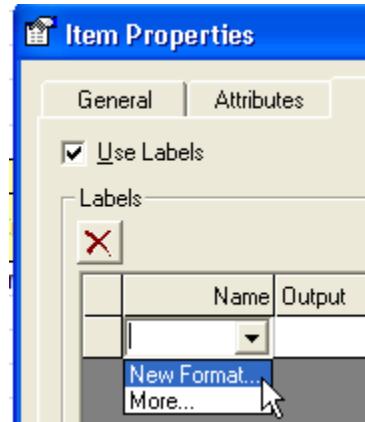
Report Formatting using New or Existing Labels

New Format

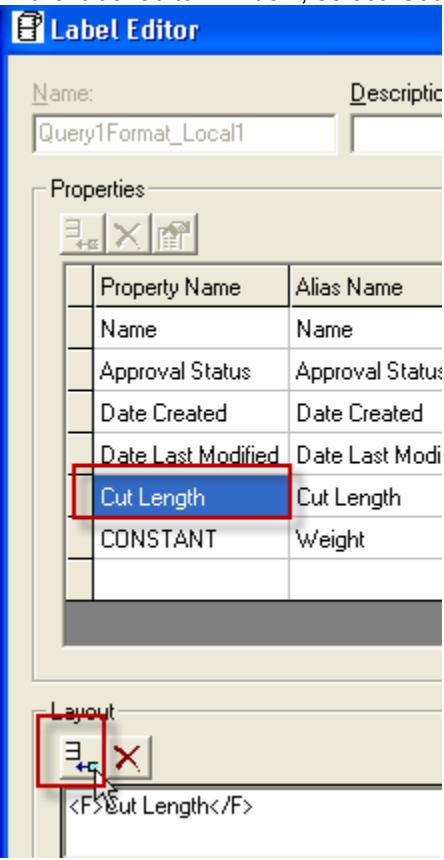
1. Right mouse click on 'Spreadsheet Reports' and select 'Create Report'
2. Expand the Reports\Examples tree and select 'MyReport1A with Embedded Label Query' and click OK. This will create a new report in the detail view.
3. Rename the report to 'MyReport1A with Formatting'
4. Right mouse click on 'MyReport1A with Formatting' and Edit Template.
5. Click the formatting tab and enter design layout.
6. Right mouse click on 'Query1' and select 'Properties'



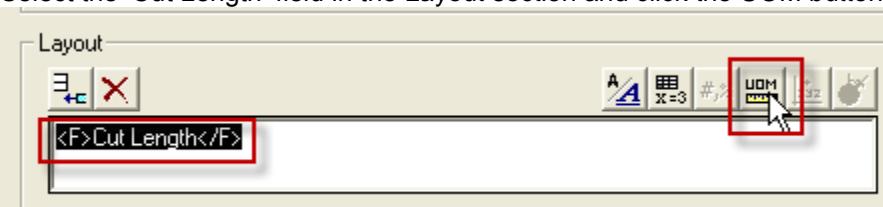
7. Click the 'Labels' tab and select the 'Use Labels' check box.
8. Click in the first empty row in the 'Name' column and select 'New Format...'. This opens the label editor.



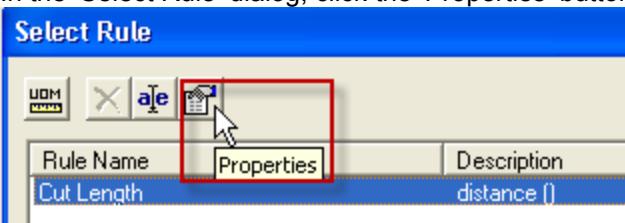
9. In the label editor window, select 'Cut Length' in the Properties section and click 'Add..'



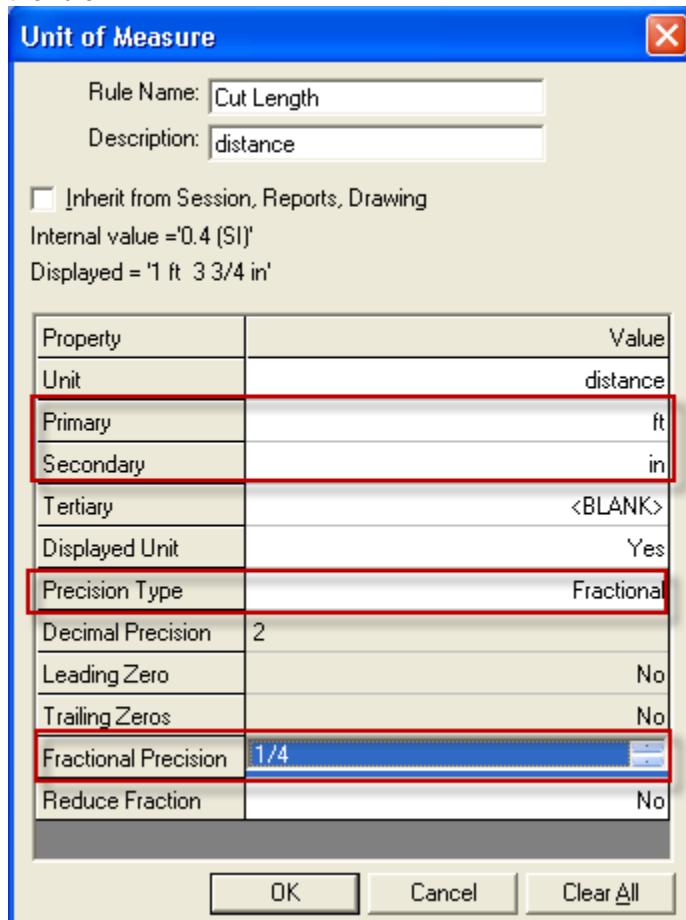
10. Select the 'Cut Length' field in the Layout section and click the UOM button



11. In the 'Select Rule' dialog, click the 'Properties' button.



12. Select values for units as ft-in, fractional with $\frac{1}{4}$ inch precision as shown and click OK to define the rule.

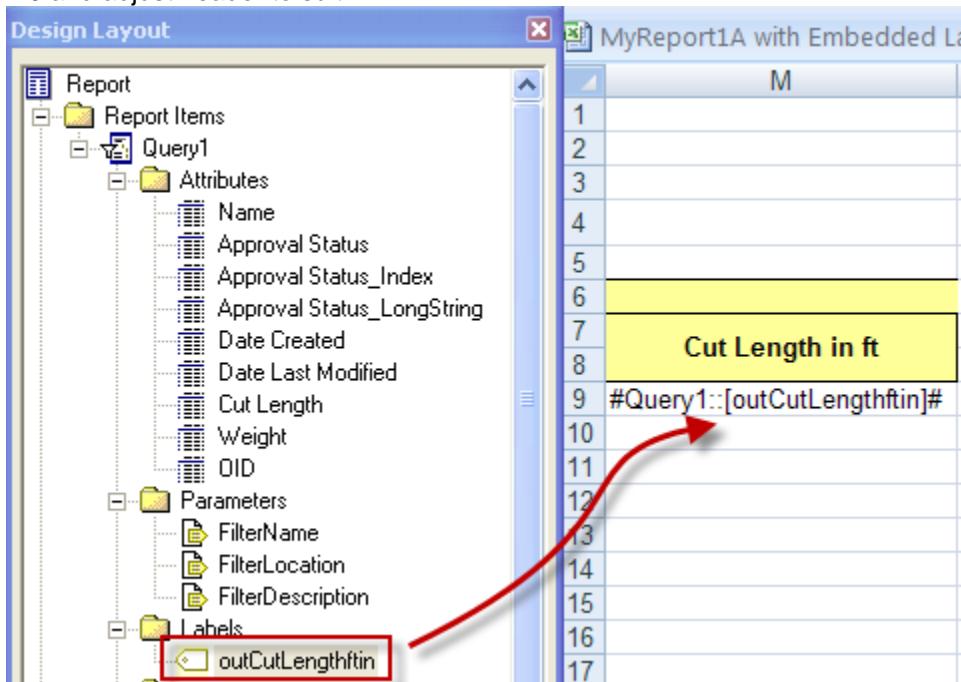


13. Click OK to select the rule
14. Click OK to define the new format.
15. Click the Output column and edit 'outQuery1Format_Local1' to 'outCutLengthftin'



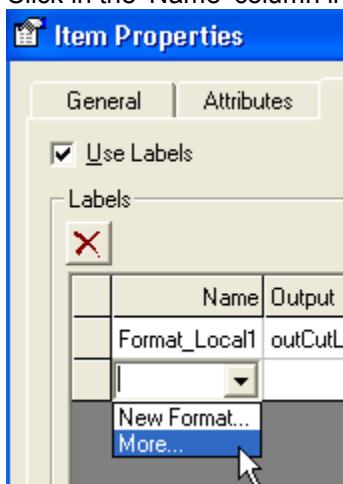
16. Click OK to define the new format.

17. A new branch 'Labels' has appeared in the tree. Drag and drop the outCutLengthftin label to cell M9 and adjust header to suit.

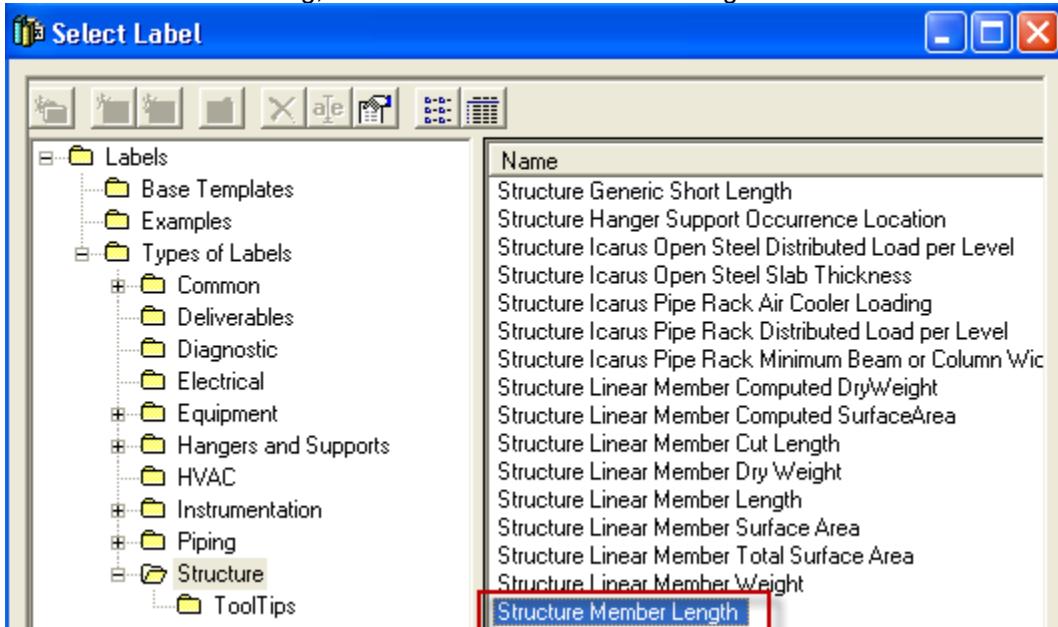


Existing Label Format

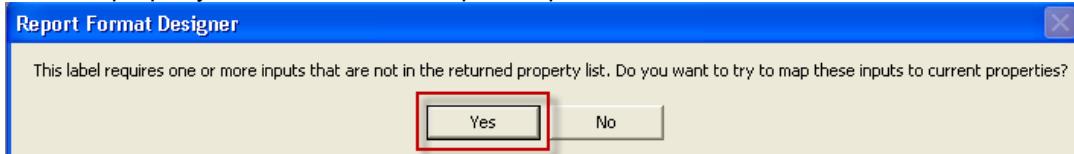
18. Right mouse click on Query1 and select Properties
19. Select the Labels tab
20. Click in the 'Name' column in the first empty row and click 'More...'



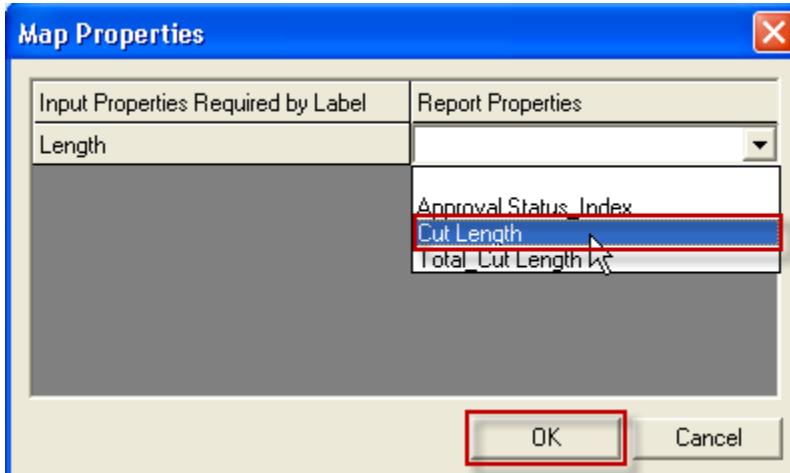
21. In the 'Select Label' dialog, select the 'Structure Member Length' label and click OK.



22. A message is displayed which informs us that the label requires an input which is not in the returned property list. Click 'Yes' to map the inputs.



23. The 'Map Properties' dialog is shown. Select 'Cut Length' in the 'Report Properties' dialog and click OK.



24. Click the Output column and edit 'outStructure Member Length_Local1' to 'outCutLengthByExistingLabel' and click OK.

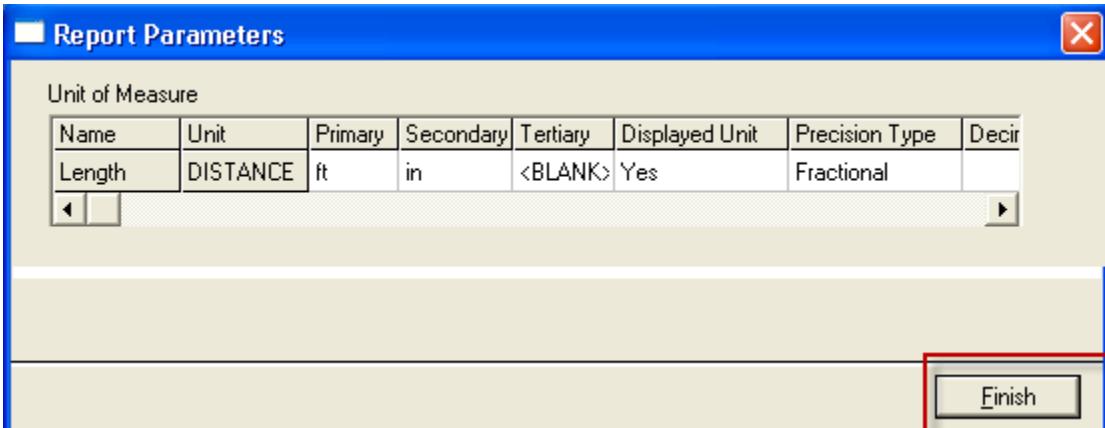
25. Drag the new label to cell N9 and edit the header to suit.

Cut Length by Existing Label	
#Query1::[outCutLengthbyExistingLabel]#	

26. Close Excel and save when prompted.

27. File → Save Report Template

28. Close the report template using the lower close box. A Report Parameters dialog is shown. Click Finish.



29. Update report and observe results –notice that while the newly created label rounds to $\frac{1}{4}$ in, the existing label rounds to $\frac{1}{2}$ in.

Cut Length in ft	Cut Length by Existing Label
12 ft 9 in	12 ft 9 in
25 ft 9 in	25 ft 9 in
25 ft 9 in	25 ft 9 in
25 ft 9 in	25 ft 9 in
14 ft 4 in	14 ft 4 in
14 ft 4 in	14 ft 4 in
14 ft 4 in	14 ft 4 in
14 ft 4 in	14 ft 4 in
25 ft 0 in	25 ft 0 in
3 ft 0 in	3 ft 0 in
3 ft 0 in	3 ft 0 in
3 ft 0 in	3 ft 0 in
3 ft 0 in	3 ft 0 in
15 ft 10 in	15 ft 10 in
13 ft 11 2/4 in	13 ft 11 1/2 in
22 ft 9 in	22 ft 9 in
19 ft 10 3/4 in	19 ft 10 1/2 in
2 ft 11 2/4 in	2 ft 11 1/2 in
11 ft 10 3/4 in	11 ft 10 1/2 in

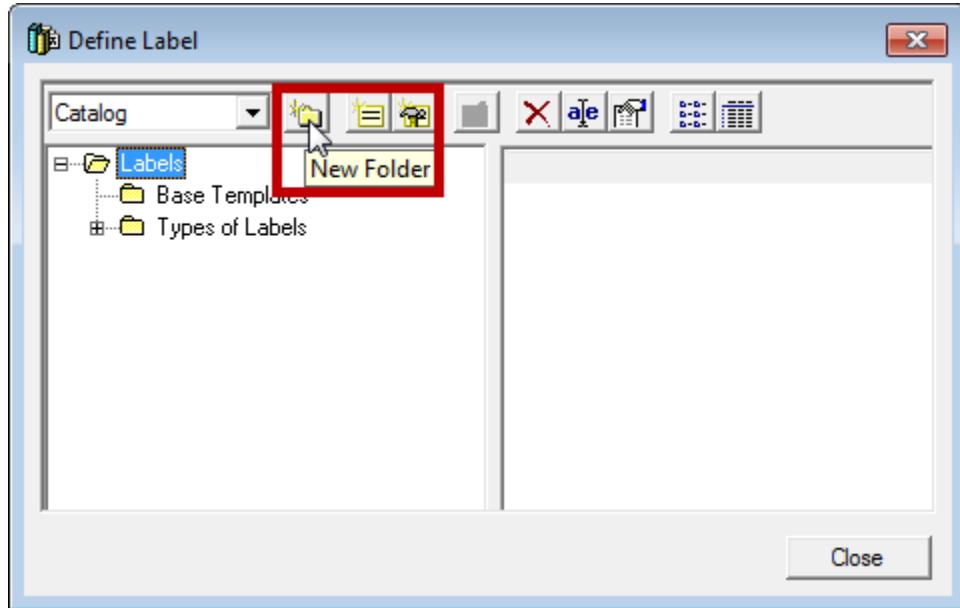
Using a PHYSICAL token with UOM

Objective: Create a label that will return the Dry Weight for Equipment. Format the Dry Weight as follows:

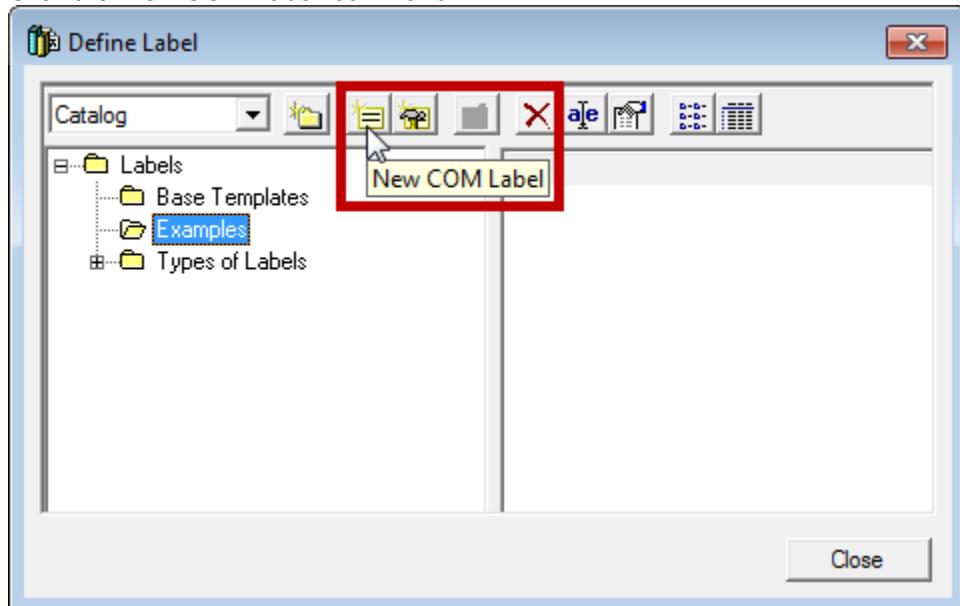
Primary UOM - kg
 Secondary UOM - g
 PrecisionType - Decimal
 DecimalPrecision – 2 places after the decimal point
 LeadingZero - None
 TrailingZeros – None

Solution:

1. Switch to the Catalog task
2. Tools → Define Label
3. Select the Labels folder and select the 'New Folder' command

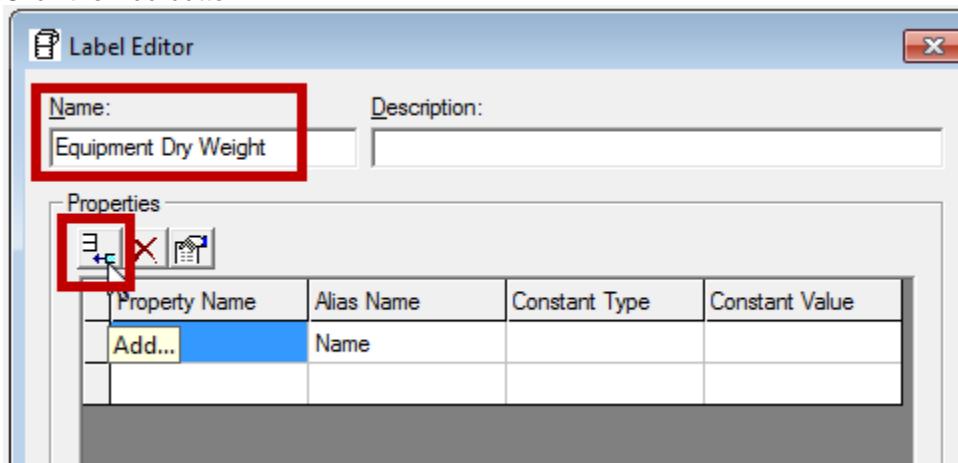


4. Name the folder 'Examples'
5. Click the 'New COM Label' command.

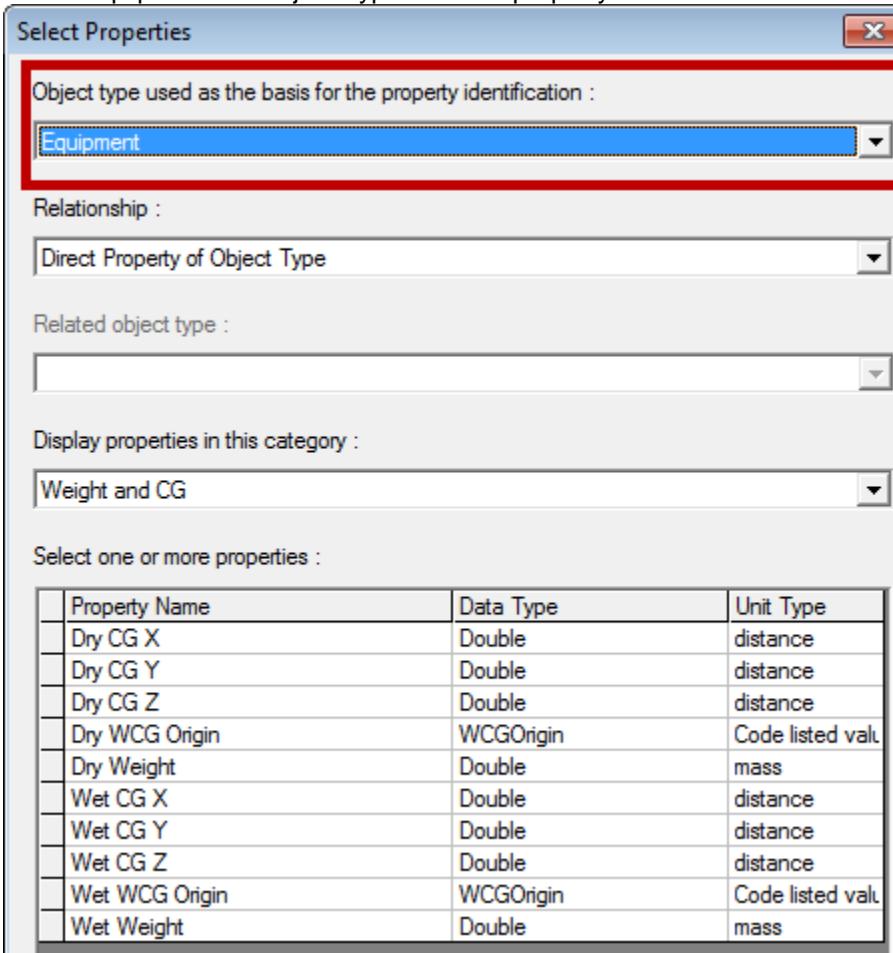


6. Name the label *Equipment Dry Weight*

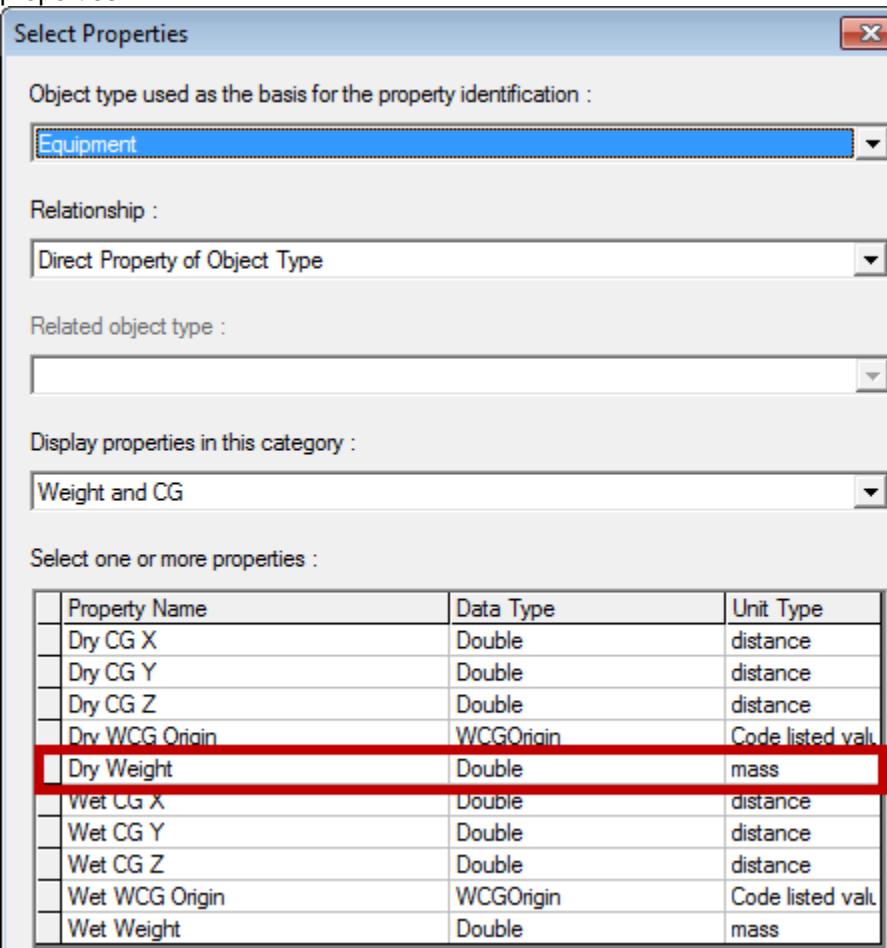
7. Click the Add button



8. Select Equipment for Object Type used for property identification

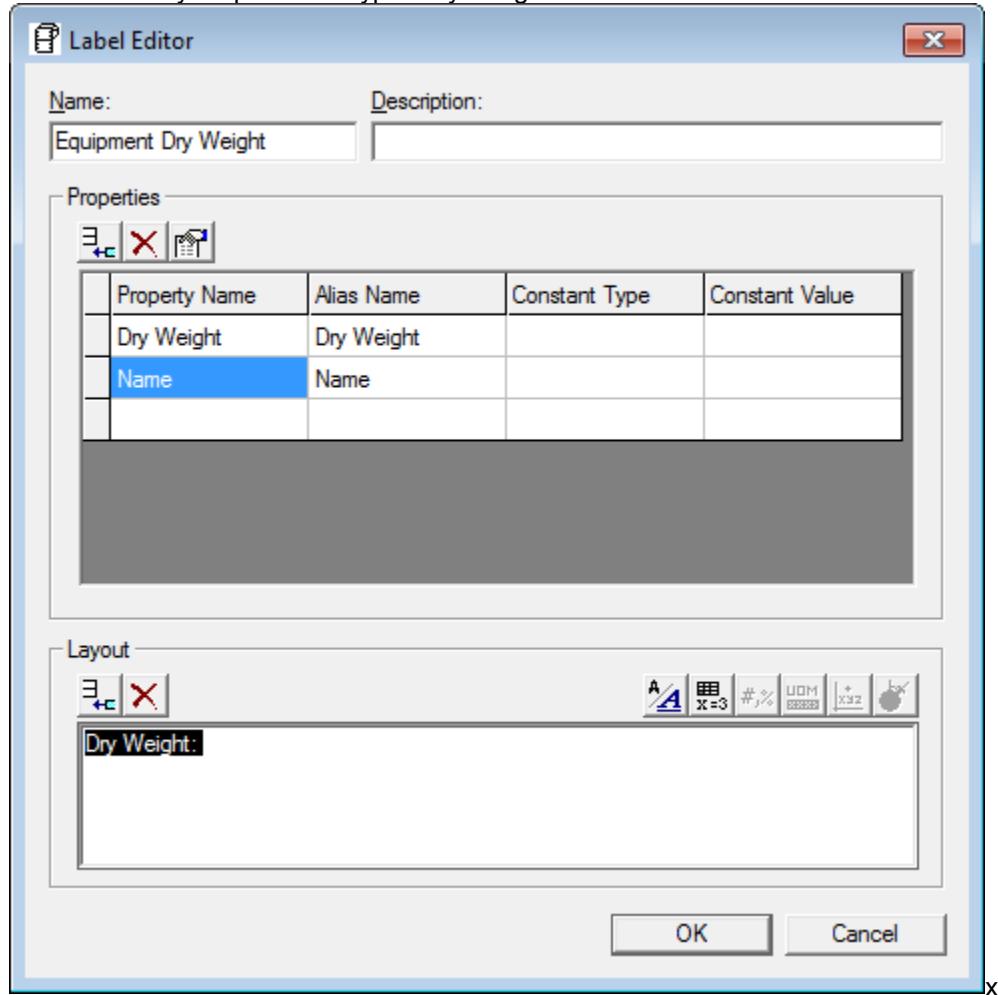


9. Select the *Dry Weight* property from the *Weight and CG* category of the Equipment direct properties.

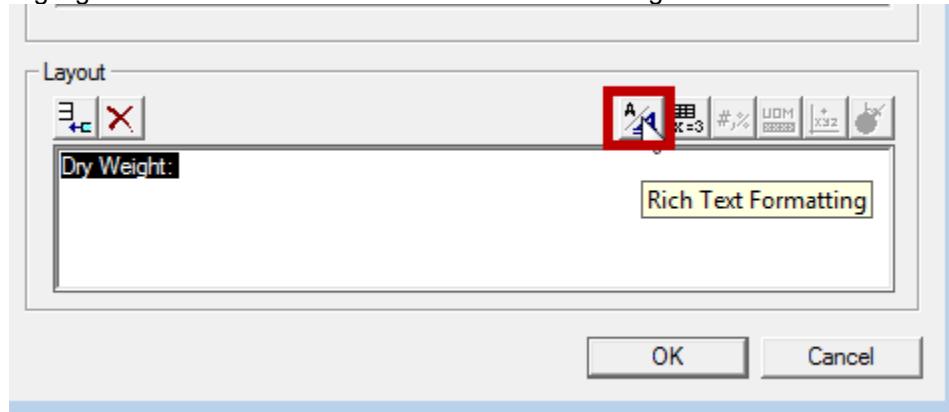


10. Click "OK". This creates a query that will extract this property. Now we need to add it to the label layout so that it can be shown, and we also need to format it.

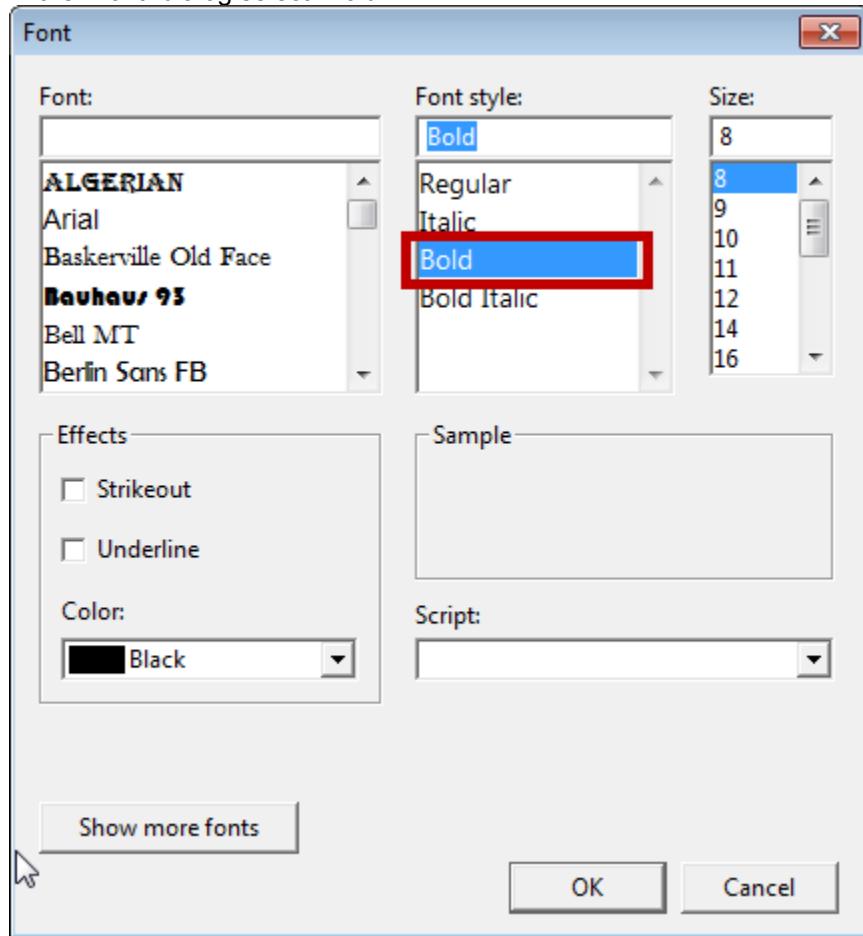
11. Click in the Layout pane and type “Dry Weight:” so that we know what the label is returning.



12. Highlight the text and click on the “Rich Text Formatting” button.

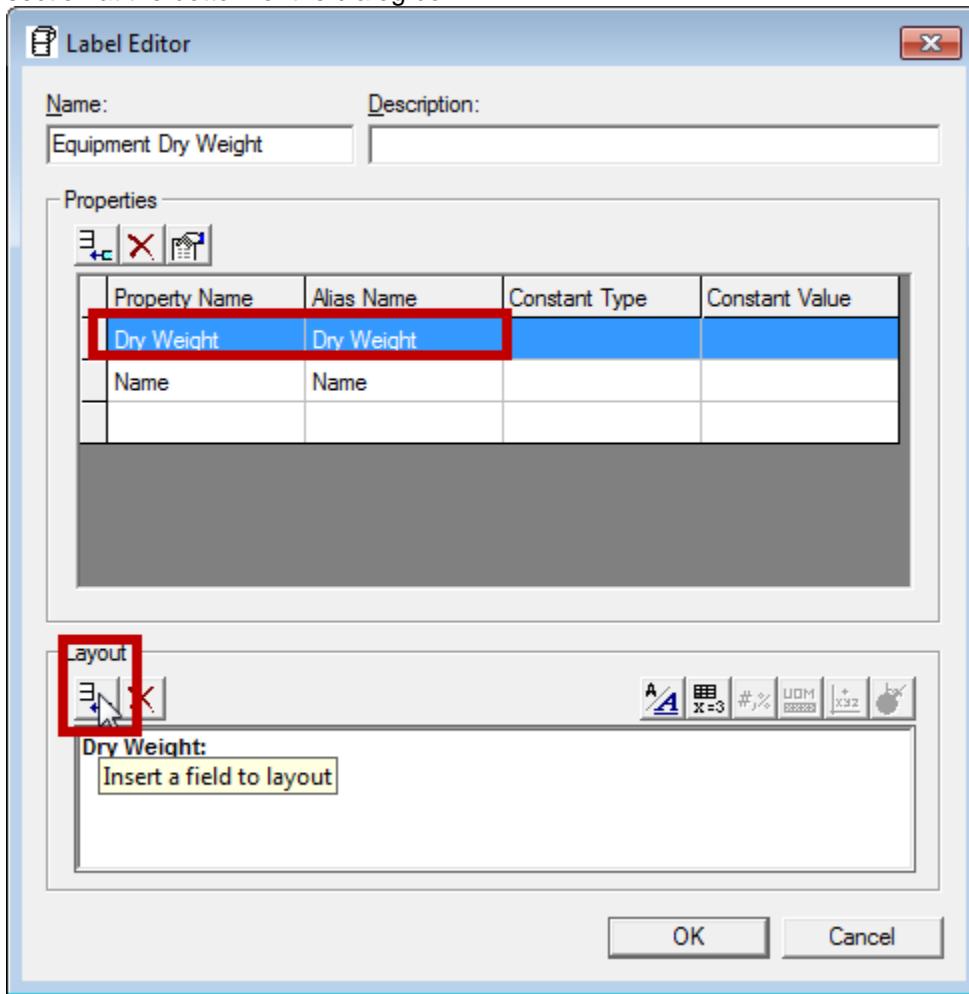


13. In the “Font” dialog select “Bold”

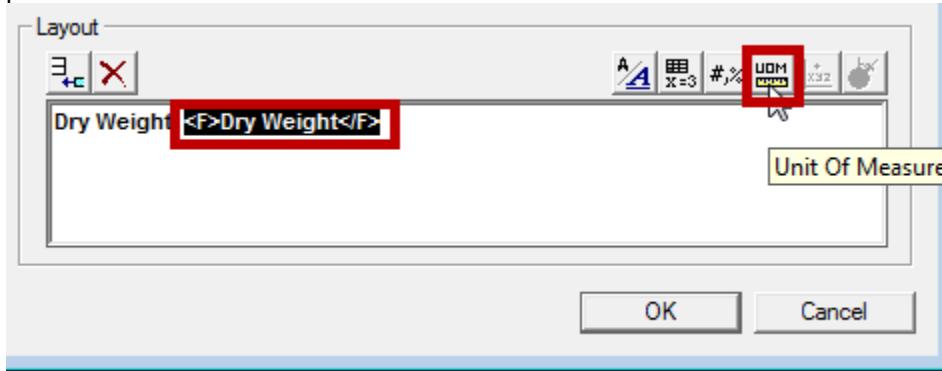


14. Click “OK”

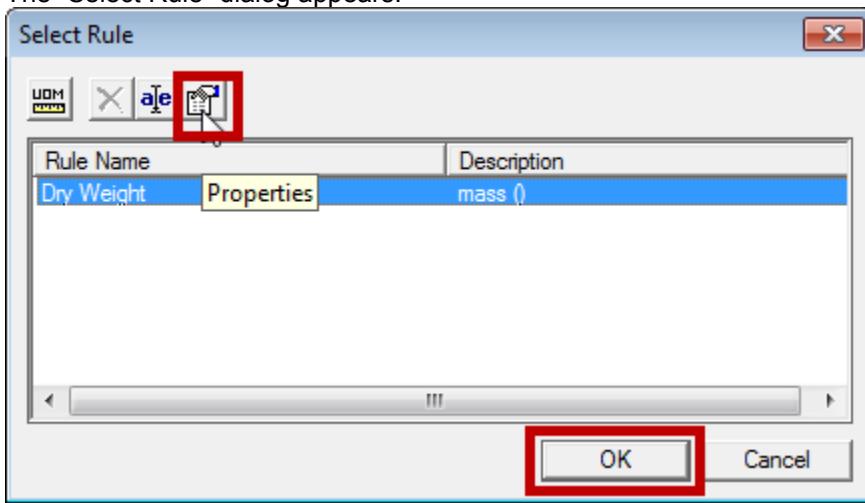
15. Select the Dry Weight property row and click “Insert a field to layout” button in the “Layout” section at the bottom of the dialog box.



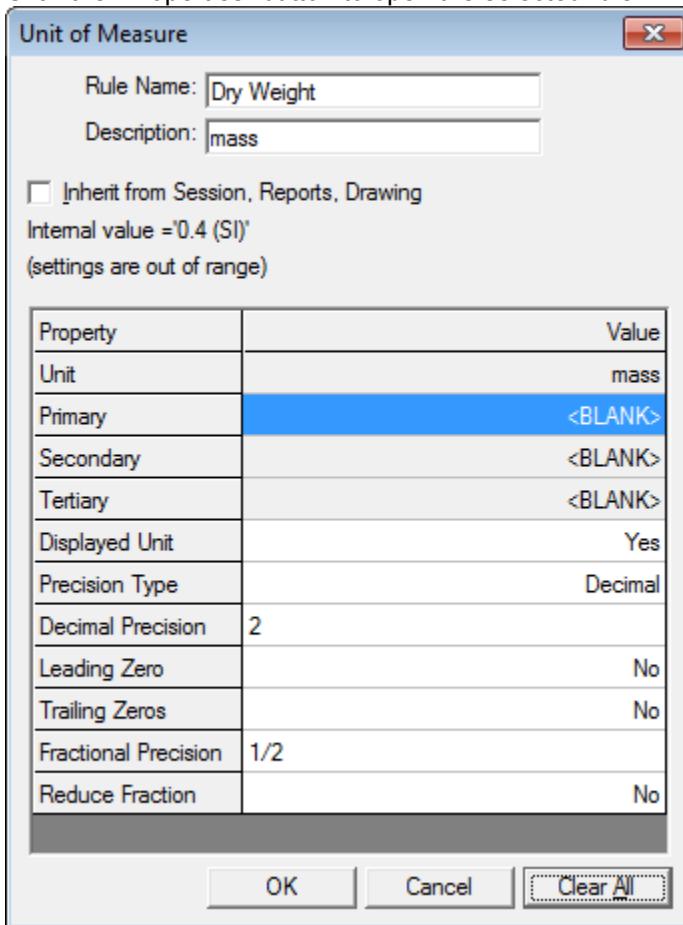
16. The text <F>Dry Weight</F> will appear in the Layout pane next to Dry Weight:. Select it, and press the “Unit of Measure” button.



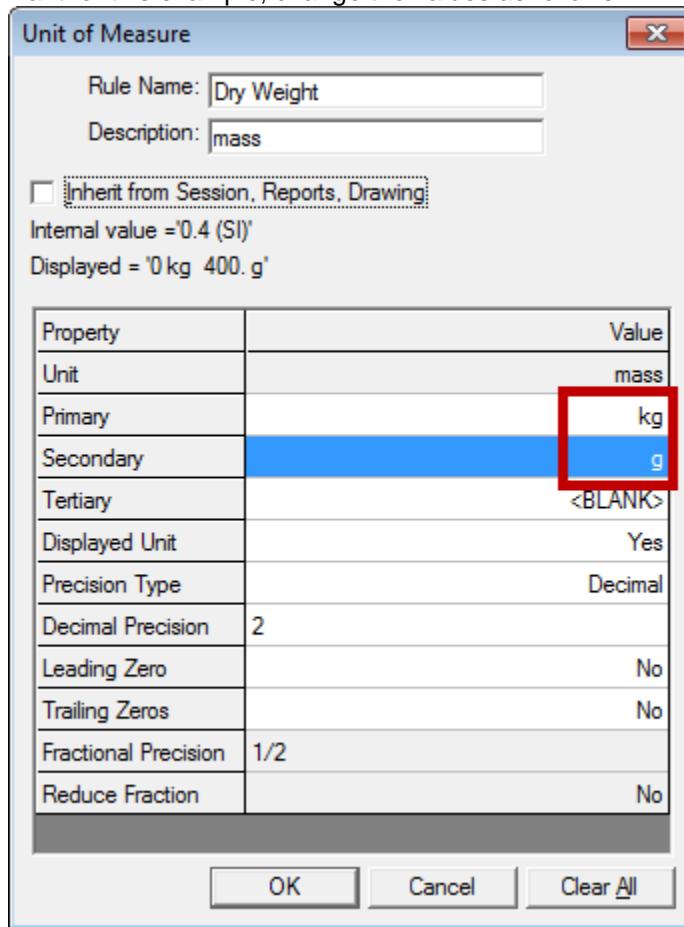
17. The “Select Rule” dialog appears.



18. Click the “Properties” button to open the selected rule

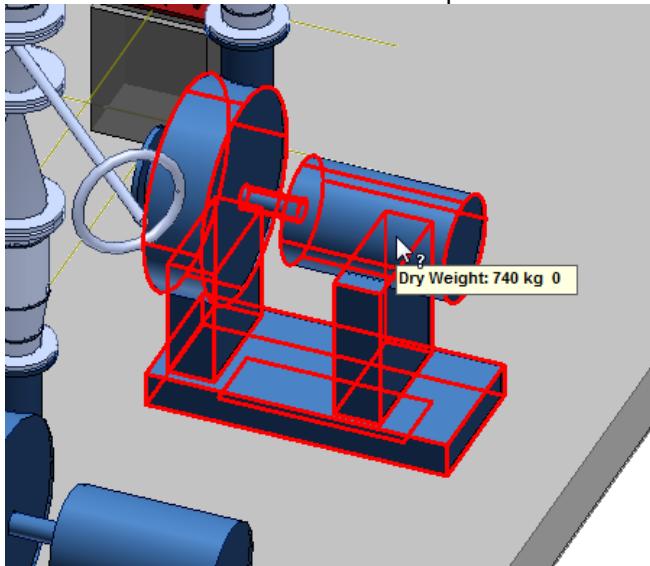


19. The “Unit of Measure” dialog controls the units used and their formatting. To get the formatting we want for this example, change the values as follows



20. Click “OK” on the “Unit of Measure” dialog. Click “OK” on the “Select Rule” dialog. Click “OK” on the “Label Editor” dialog.

21. You can now test the label as a tool tip:



Now let's open the Equipment Dry Weight.rfm file and examine it.

```

<?xml version="1.0" encoding="UTF-8"?>
<REPORT_FORMATTING
  Name="Equipment Dry weight"
  Description="">
  <DESIGN_TIME>
    Progid="SP3DLabelFormatDesigner.RTFLabel"
    Action=""
    Arg=""/>
  <RUN_TIME>
    Progid="SP3DLabelsFormat.FormatLabel"
    Action="RTFLabel"
    Arg=""/>
  <FORMATTING_PARAMETERS>
    Name="Equipment Dry weight"
    Site="User"
    Path="Equipment Dry weight.rfp" />
  <LAYOUT_TEMPLATE>
    Type="Internal" />
  <RTF_LABEL>
    <POINTS />
    <VECTORS />
  <BLOCKS>
    <BLOCK
      Action="visible"
      ChildId="No">
      <TOKENS>
        <TEXT
          Value="{\rtf1\ansi\deff0{\fonttbl{\f0\fswiss\fcharset0 Arial;}{\f1\fni1\fcharset0 MS Shell Dlg;}}{\colortbl ;\red0\green0\blue0;}\viewkind4\uc1\pard\cf1\lang1033\b\f0\fs16 Dry weight: "
          Toparse="no"
          visible="yes" />
        <PHYSICAL
          Column="dry weight"
          UOM="dry weight"
          visible="yes" />
        <TEXT
          Value="\cf0\b0\f1\fs17 \par " "
          Toparse="no"
          visible="yes" />
      </TOKENS>
    </BLOCK>
  </BLOCKS>
</RTF_LABEL>
</REPORT_FORMATTING>

```

The formatting here happens in the <BLOCKS> section. This section is made of one BLOCK (more on that later), and each BLOCK consists of TOKENS.

TEXT – user defined (constant) text formatted as Rich Text. The *Value* attribute shows the actual text string together with the Rich Text formatting. In our case, the user text is “Dry Weight”. The Rich Text formatting itself is rather cryptic, but all you need to know is that it applies to all tokens up until the last closing “}” (in our case: *Value="\\cf0\\b0\\f1\\fs17 \\par }*). That means that it will be applied to the data returned from the PHYSICAL token too:

PHYSICAL – describes dimensioning data that will come from the database and will require transformation to some units of measure. In our case this is the Dry Weight. The *Column* attribute describes the name of the property

(<RETURNED_PROPERTY Name="Dry Weight"> from the .rqe file, or the alias from the SQL statement if we have an SQL label). The attribute *UOM="Dry Weight"* tells the software how to format the Dry Weight value. The specifics of the UOM formatting are found in the Equipment Dry Weight.rfp file:

```
<?xml version="1.0" encoding="UTF-8"?>
<FORMATTING_PARAMETERS
  Name="Equipment Dry weight"
  Description="">
  <DESIGN_TIME
    Progid=""
    Action=""
    Arg="" />
  <RUN_TIME
    Progid="SP3DPrompts.ct1Tab"
    Action=""
    Arg="" />
  <FIELDS />
  <UOMS>
    <UOM
      Name="Dry weight"
      CanInherit="No"
      Type="mass"
      Primary="kg"
      Secondary="g"
      Tertiary=""
      UnitsDisplayed="Yes"
      PrecisionType="Decimal"
      DecimalPrecision="2"
      FractionalPrecision="2"
      LeadingZero="No"
      TrailingZeros="No"
      ReduceFraction="No" />
  </UOMS>
</FORMATTING_PARAMETERS>
```

As you can see, the <UOMS> just repeat the values we set in the Unit of Measure dialog and are pretty straightforward. The one important attribute to remember here is *CanInherit* – it determines if the UOM's can be overwritten by the UOM of the calling application. This means that if you set this attribute to "Yes", even if you select Metric UOM, you'll still see your values expressed in the English system if that's what you have selected in your session file.

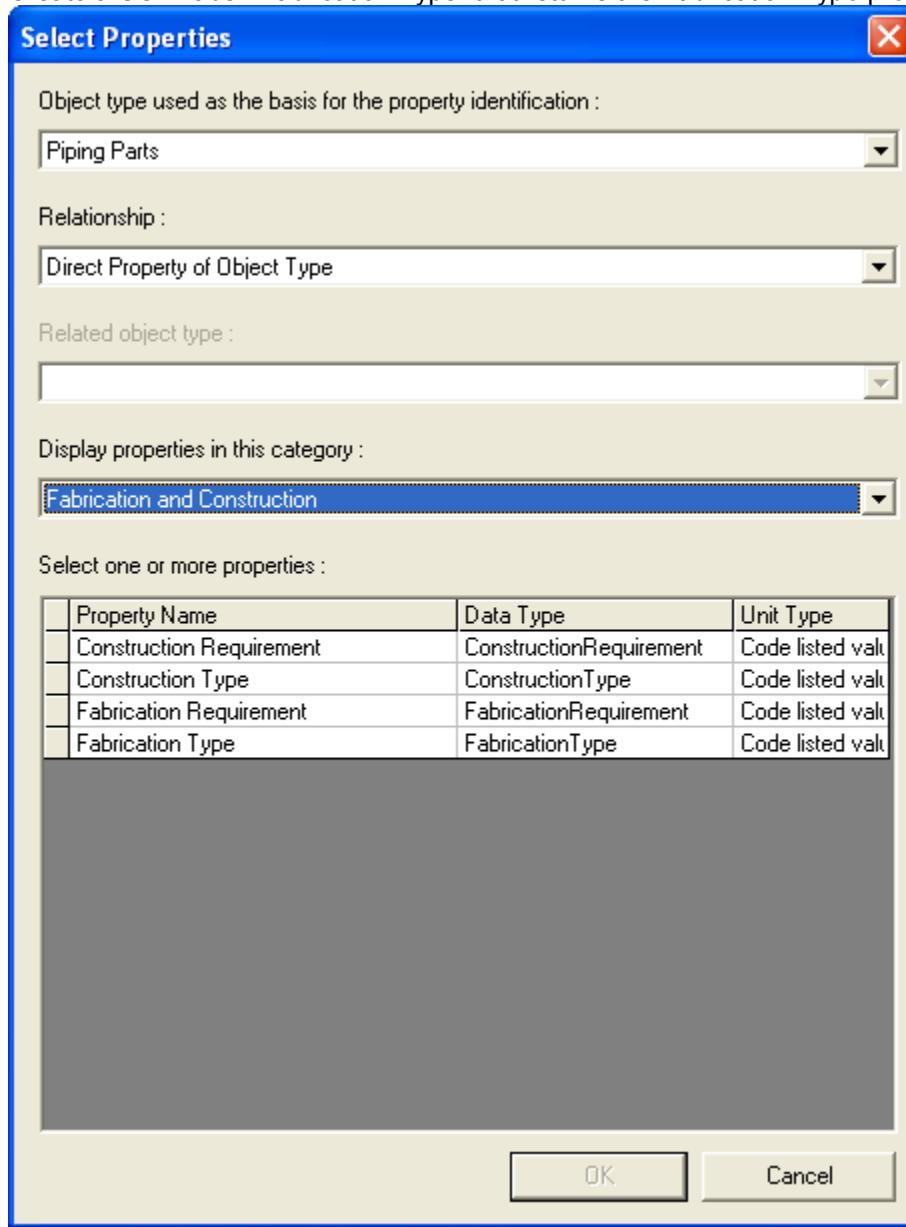
It is not necessary to remember all the attributes of the <UOM> tag, but it's good to have a general idea of what they are and where they located. Probably the only times you'll ever need to create one of those "by hand" will be if you are creating an SQL or VB label, but even then you can just copy them from another label.

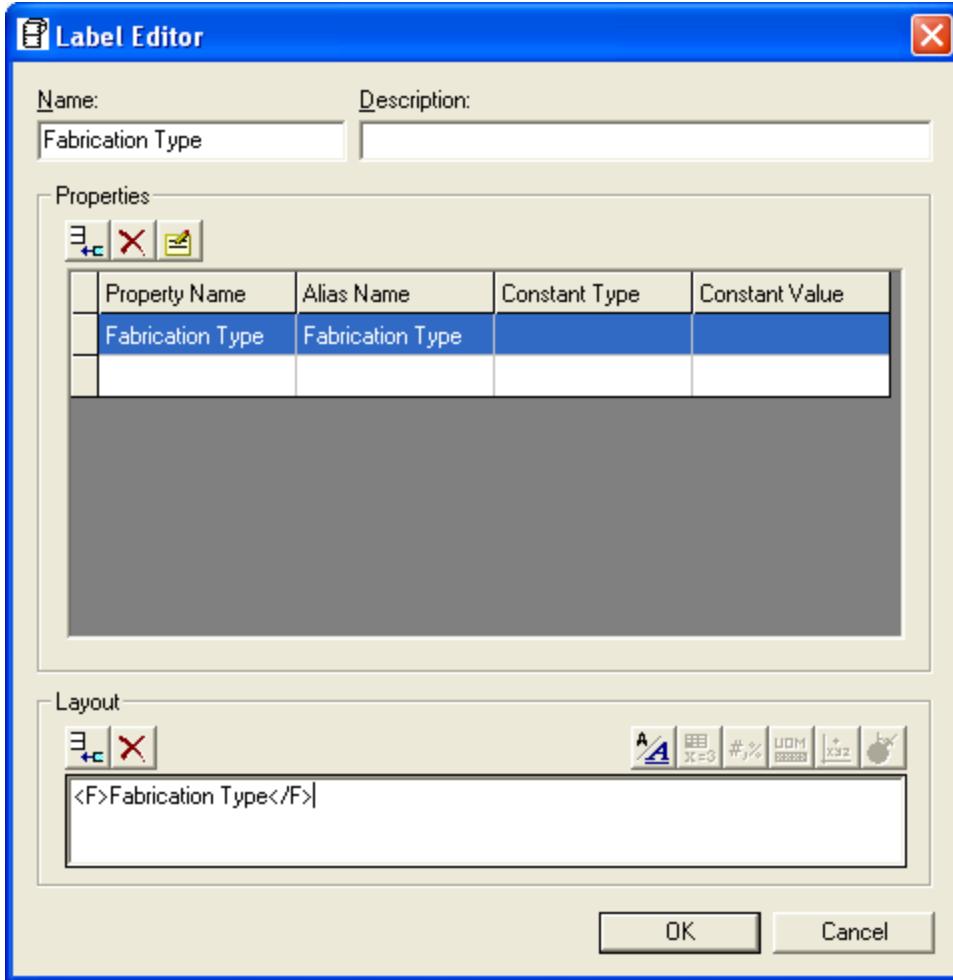
Outputting codelisted values

Objective: Create a label for Piping Parts that returns the Code List value for Fabrication Type as Number, Short String Value and Long String Value

Solution:

1. Create a COM label "Fabrication Type" that returns the Fabrication Type property as shown:





2. The so created label will return the Short Description, by default:
3. Open the .rfm file.

```
<?xml version="1.0" encoding="UTF-8"?>
<REPORT_FORMATTING
  Name="Fabrication Type"
  Description="">
  <DESIGN_TIME
    ProgId="SP3DLabelFormatDesigner.RTFLLabel"
    Action=""
    Arg="" />

  <RUN_TIME
    ProgId="SP3DLabelsFormat.FormatLabel"
    Action="RTFLLabel"
    Arg="" />

  <FORMATTING_PARAMETERS
    Name="Fabrication Type"
    Site="User"
    Path="Fabrication Type.rfp" />

  <LAYOUT_TEMPLATE
    Type="Internal" />

  <RTF_LABEL>
    <POINTS />

    <VECTORS />

    <BLOCKS>
      <BLOCK
        Action="visible"
        Child="No">
        <TOKENS>
          <TEXT
            Value="\rtf1\ansi\deff0{\fonttbl{\f0
              ToParse="no"
              visible="yes" />

            <DATA
              Column="Fabrication Type"
              ToParse="yes"
              visible="yes" />

            <TEXT
              value=" \par } "
              ToParse="no"
              visible="yes" />
          
        
```

4. To make the label return the numeric Code List value, change the highlighted text to:

```
<DATA
  Column="Fabrication Type_Index"
  ToParse="yes"
  visible="yes" />
```

5. If we want the label to return the long string value instead, change the line to

```
<DATA
  Column="Fabrication Type_Longstring"
  ToParse="yes"
  visible="yes" />
```

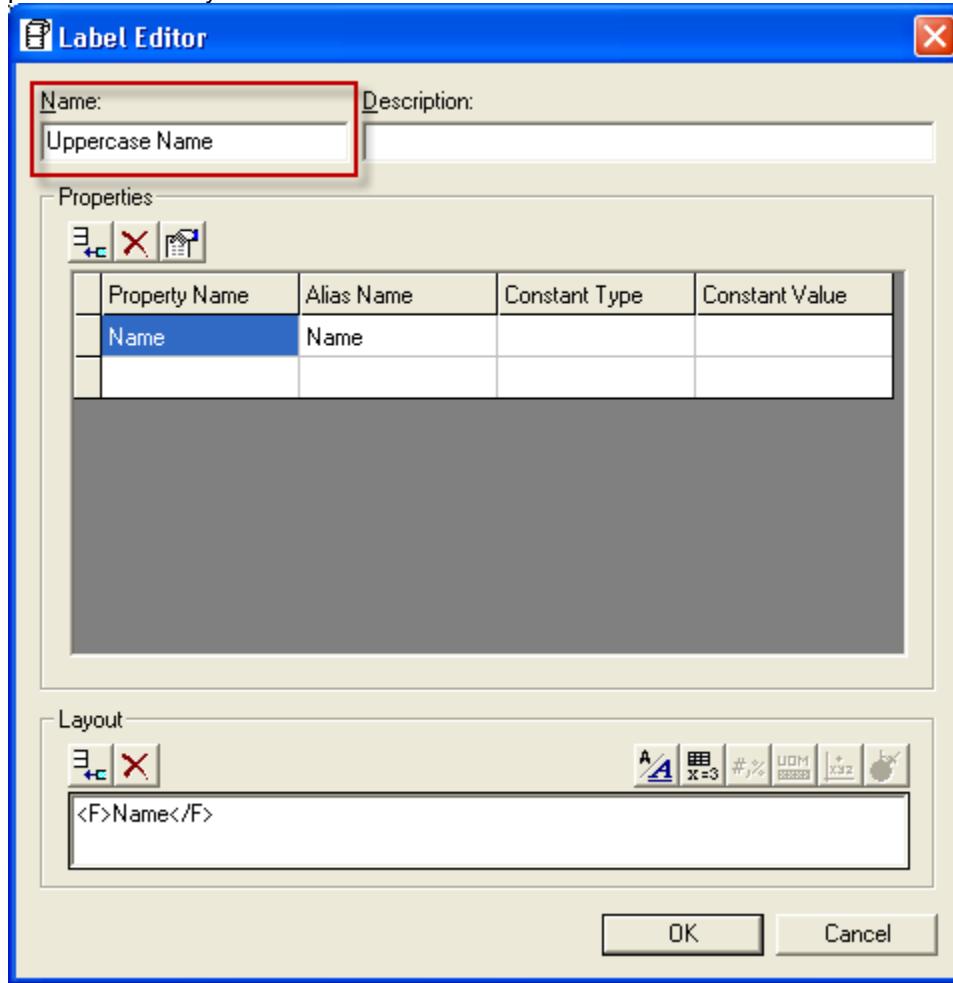
6. Test the label as a tooltip

Using a FORMAT_EXPRESSION

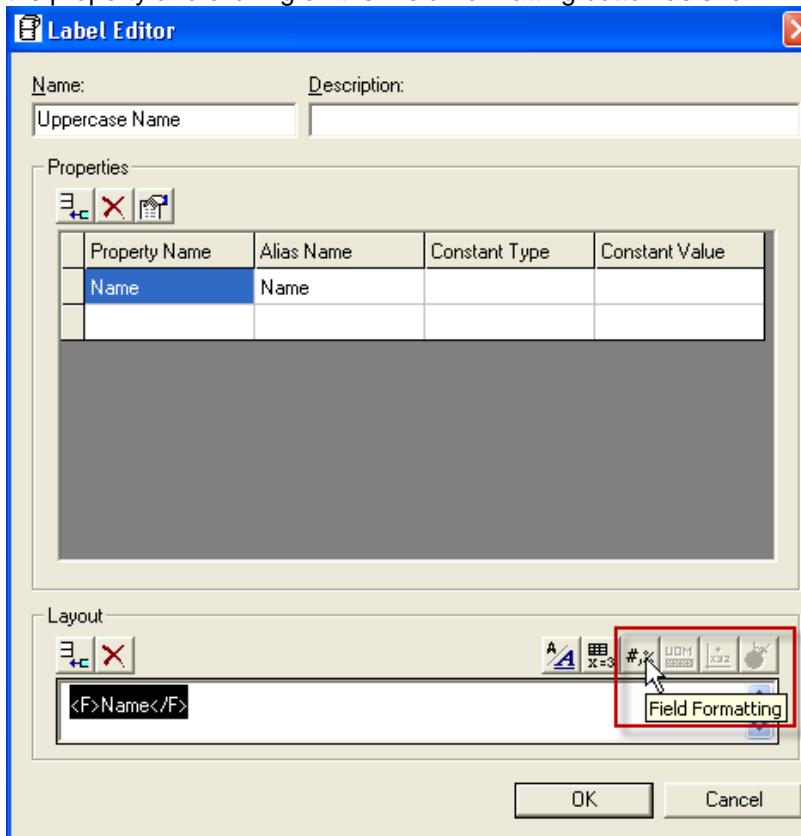
Objective: Create a label that will return a system's name all in UPPER CASE

Solution:

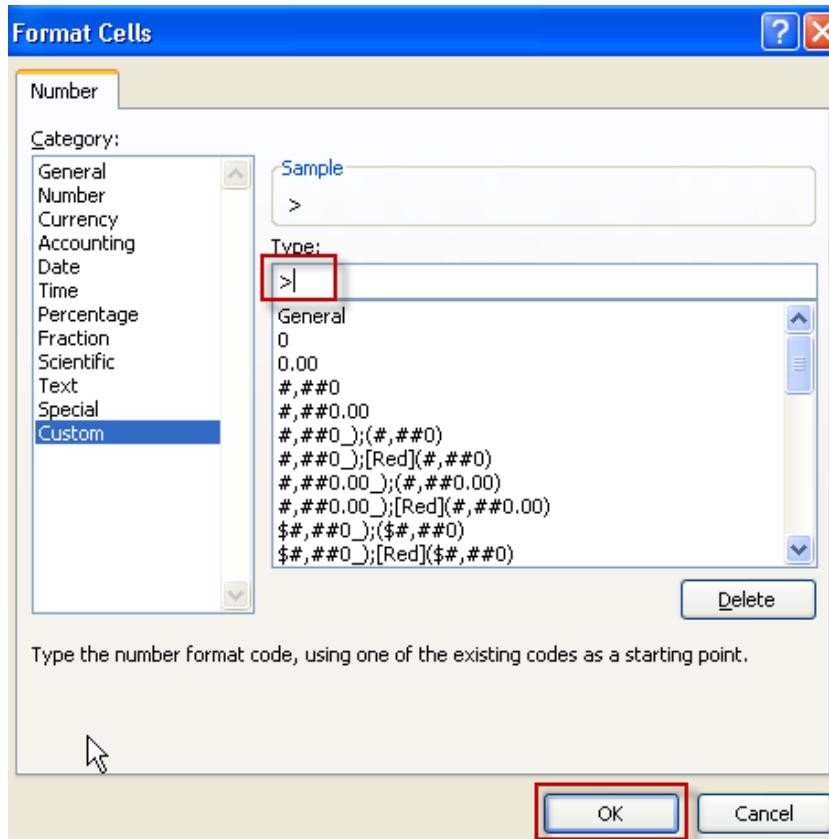
1. We can apply VB style formatting to the label. Let's see how this is done.
2. Create a label called 'Uppercase Name'. You could just use the default label with which you are presented when you create a new label:



3. Note that you can also apply a rich set of formatting options through the UI by simply highlighting the property and clicking on the Field Formatting button as shown in the picture.



- Enter the text > in the Custom box and click OK.



- Open the .rfm file and look at the TOKEN <DATA>

```
<TOKENS>
<TEXT
  value="{\rtf1\ansi\deff0{\fc
  ToParse="no"
  visible="yes" />

<DATA
  Column="Name"
  ToParse="yes"
  visible="yes">
  <FORMAT_EXPRESSION
    Expression="&gt;">
    FirstDayOfWeek=1
    FirstWeekOfYear="1" />
</DATA>

<TEXT
  value=" \par } "
  ToParse="no"
  visible="yes" />
</TOKENS>
```

- Note that even though we entered ">", but because we are using this from within XML, it was written out as <FORMAT_EXPRESSION Expression=">"/>
- Test the label as a tooltip on an object which has a mixed case name, e.g. a pipe component.

Conditional Formatting

Raw Values

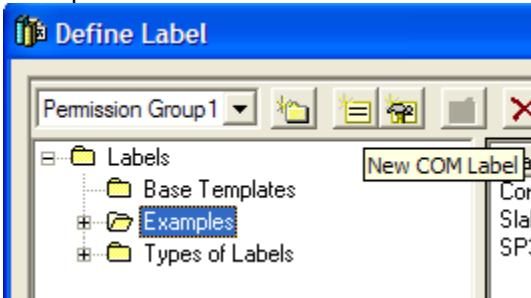
We will create a label that returns a value of High, Medium or Low based on the unit weight (weight per unit length) of a steel member.

The criteria are:

- Unit Weight > 60 High
- Between 20 and 60 Medium
- Less than 20 Low

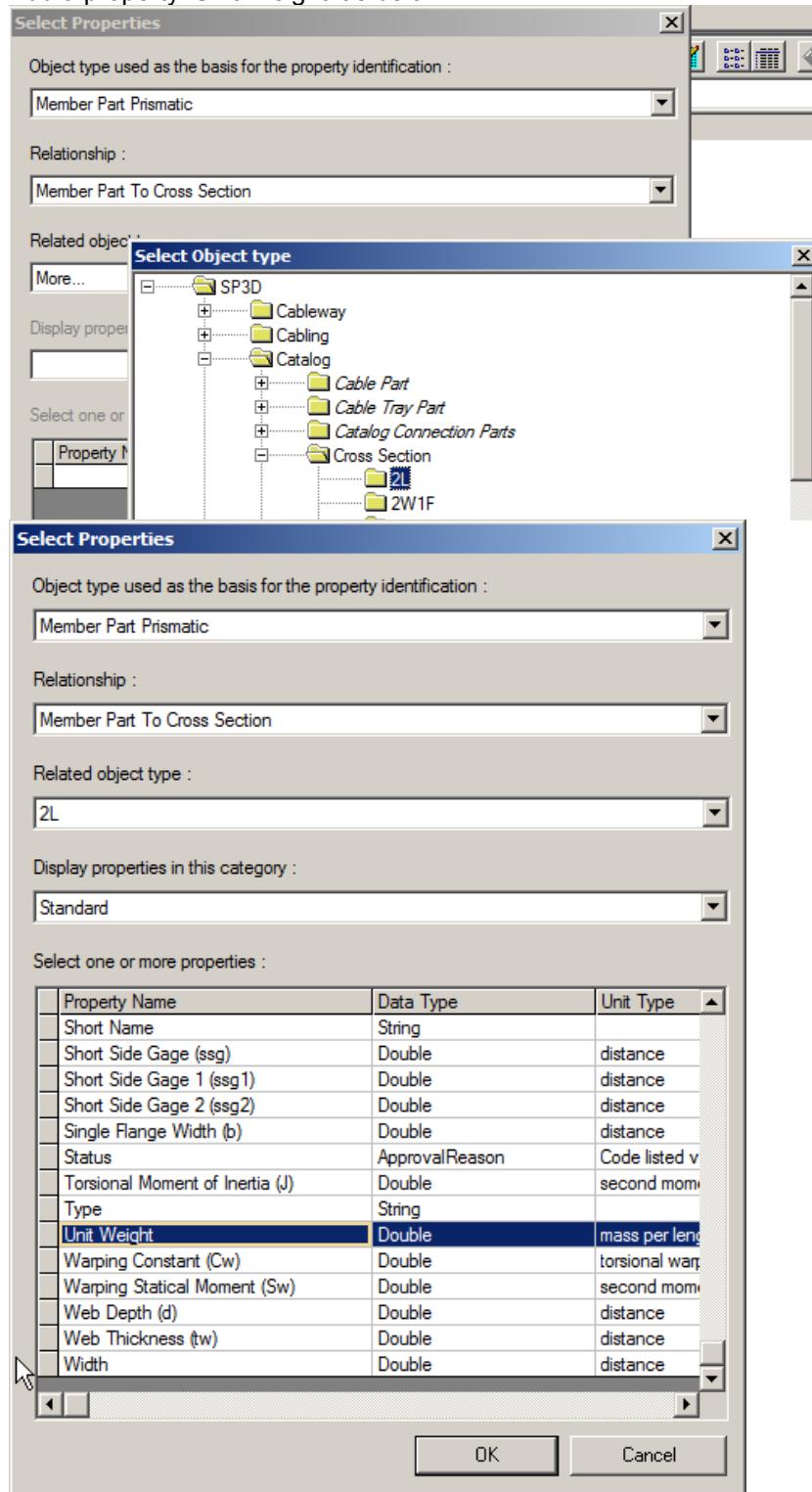
Based on the raw values from the database.

1. Switch to Catalog task
2. Tools → Define Label
3. Define a new COM label using the New COM Label command in the catalog task under the Examples folder

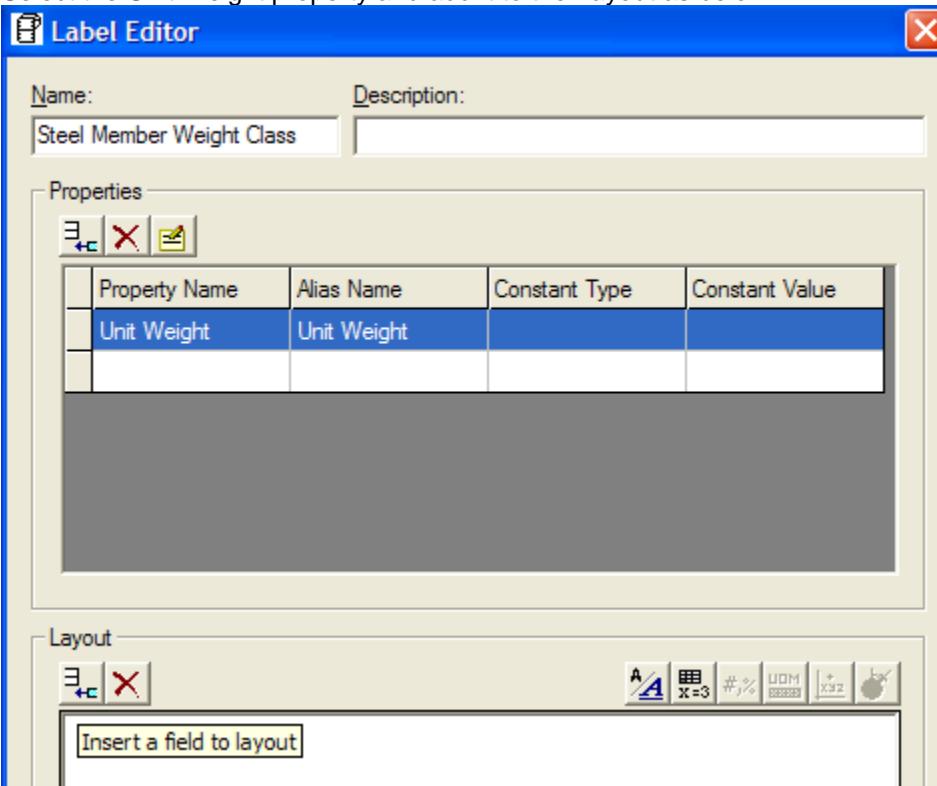


4. Name the label 'Steel Member Weight Class'
5. Remove the existing property 'Name'

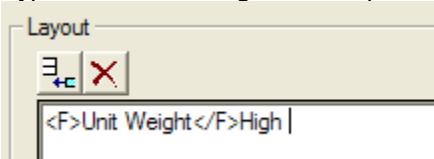
6. Add a property 'Unit Weight' as below



7. Select the Unit Weight property and add it to the Layout as below



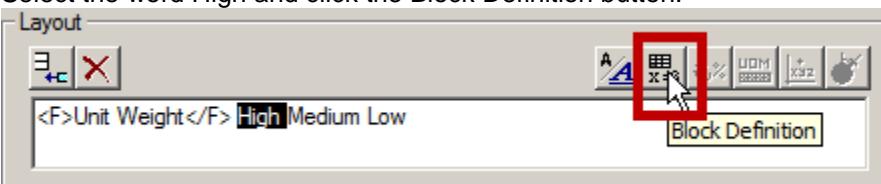
8. Type in the word 'High' and a space after it.



9. Type the word 'Medium' and a space after it.

10. Type the word 'Low' and a space after it

11. Select the word High and click the Block Definition button.



12. In the lower grid, pick the property Unit Weight

13. Select the operator ' >' and the value 60

Property	Operator	Value	State of value	Value 2
Unit Weight	>	60	Raw	

14. Select the word Medium, click the Block Definition button and fill in the definition as below.

Property	Operator	Value	State of value	Value 2
Unit Weight	Between	20	Raw	60

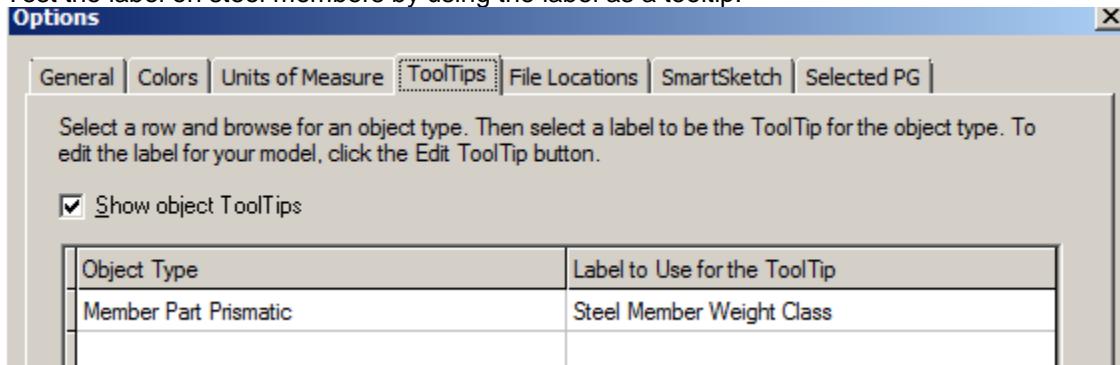
15. Select the word Low, click the Block Definition button and fill in the definition as below

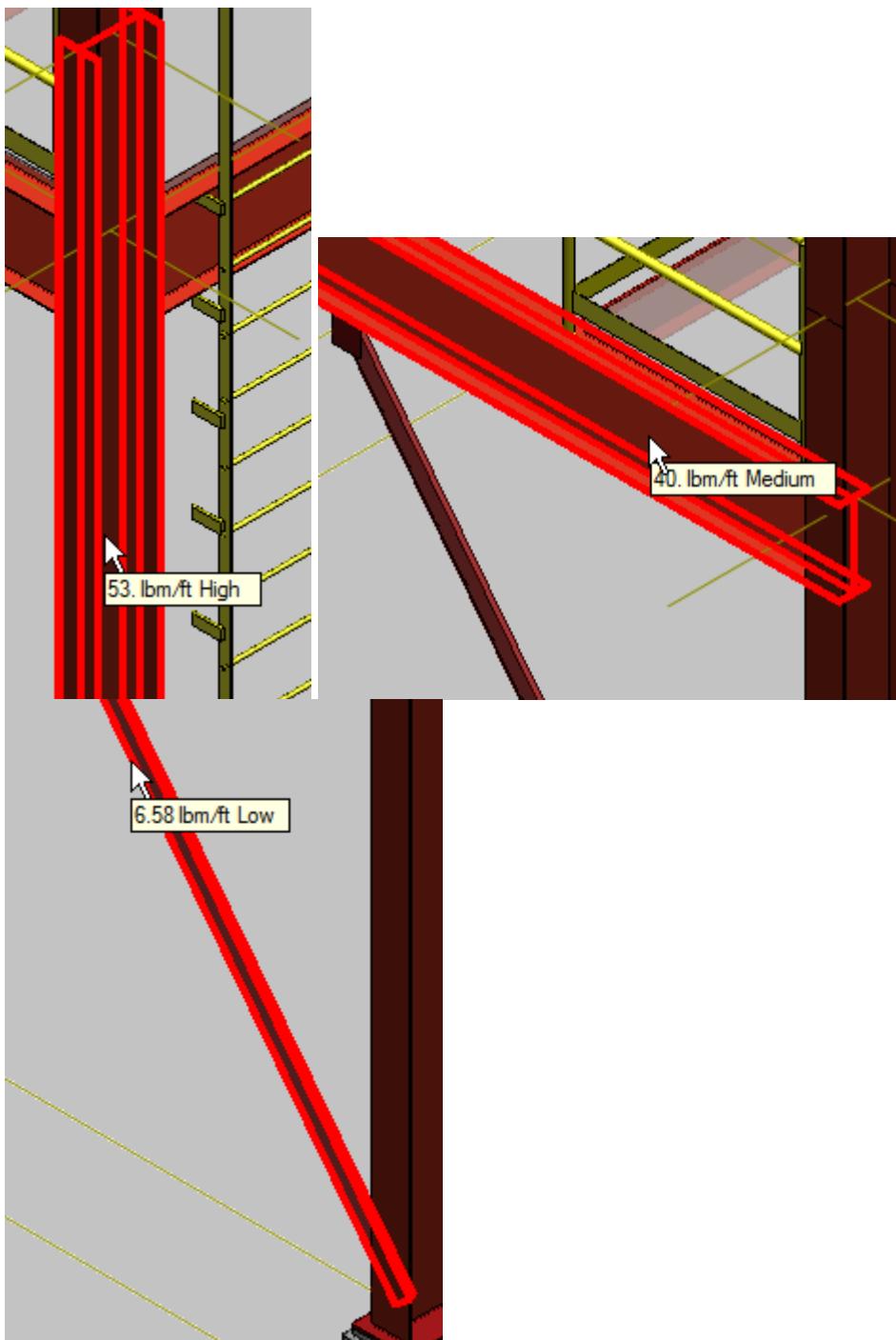
Property	Operator	Value	State of value	Value 2
Unit Weight	<	20	Raw	

16. Once the three blocks are done, click OK to complete the label.

17. Remember that all values in the database area stored in SI units. In the case of a unit weight property (weight/length), this would be kg/m. So when we compare raw values, we are actually checking for a raw value of 60 kg/m (approx. 40.3 kg/ft). In this exercise, we will check the fact that we do get three different values of High, Medium and Low for three separate members and then in the next part of this exercise we will convert the units too.

18. Test the label on steel members by using the label as a tooltip.

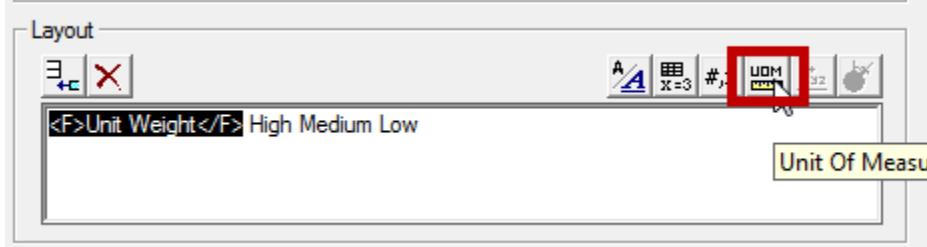




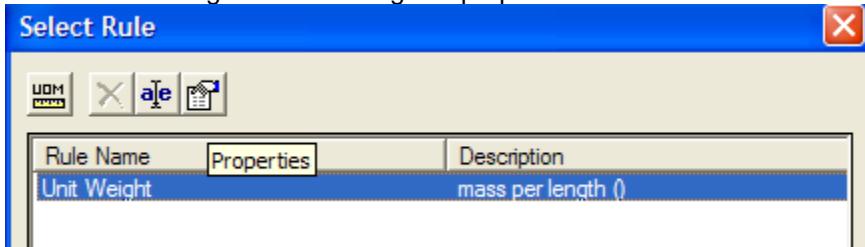
Formatted Values

We will now modify above label to operate based on formatted values of the unit weight.

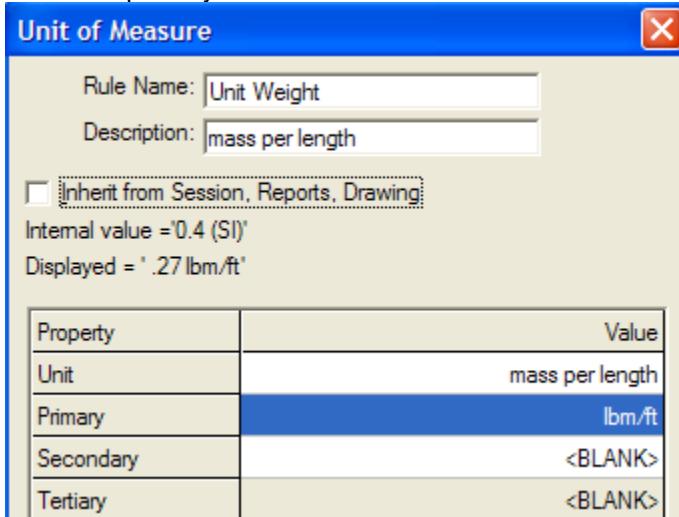
1. Edit the label we just created and select the Unit Weight property and pick the UOM button



2. Select the existing rule and change its properties



3. Choose a primary unit of lbm/ft and click OK



4. Click OK again to accept the unit.

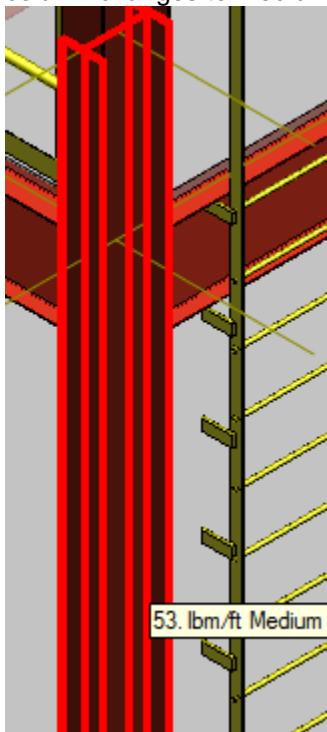
5. Select the 'High' block and change the state of value to post-formatted.

Property	Operator	Value	State of value	Value 2
Unit Weight	>	60	Post-formatted	

6. Similarly change the state of value for the other two blocks.

7. Click OK to save the label.

8. Switch to the Structure task and test the label again as a tooltip. Notice that the output for the column changes to Medium since we are now testing for formatted values



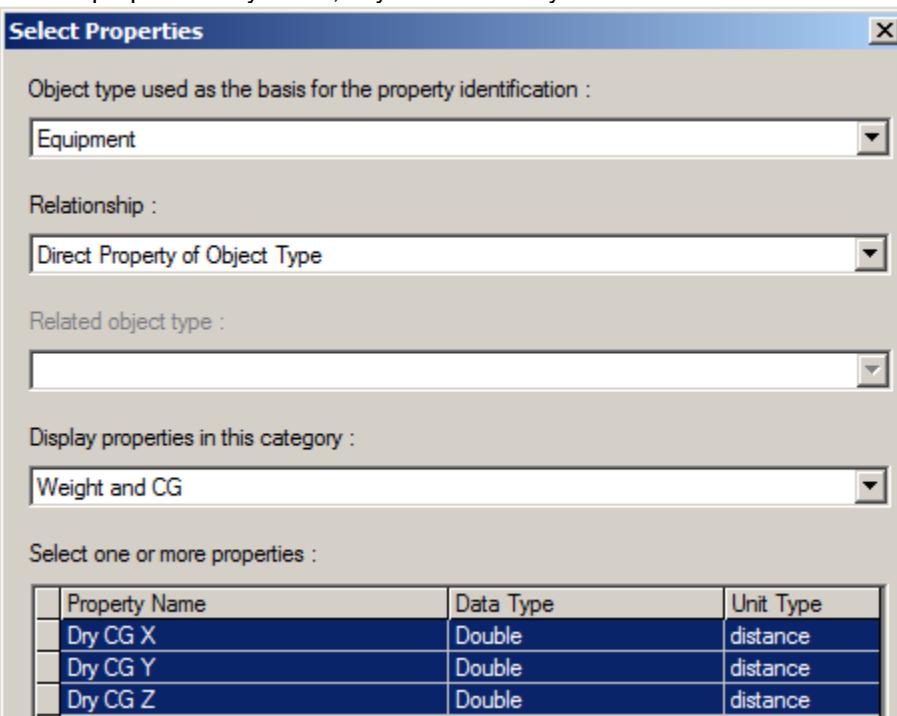
Creating a coordinate label

Objective: After this lab, you will be able to format coordinates for both unit of measure and coordinate system for reporting

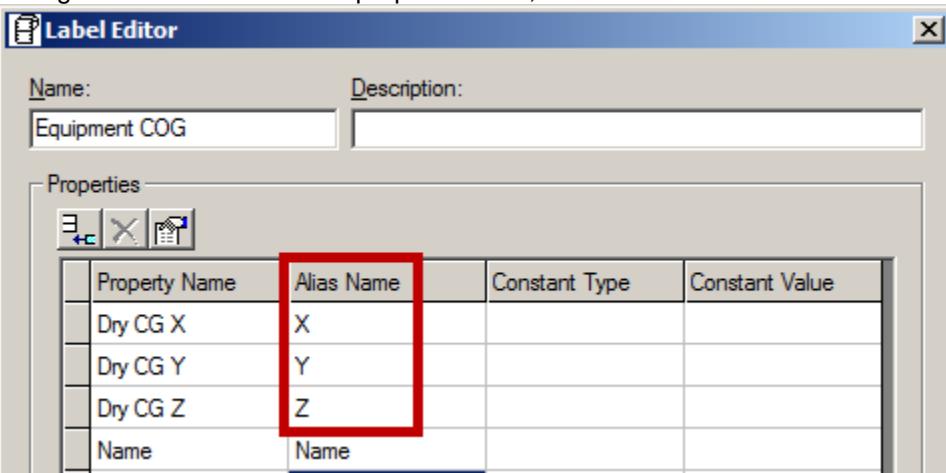
Creating the label

1. Switch to Catalog task
2. Tools → Define Label
3. Define a new COM label using the New COM Label command in the catalog task under the Examples folder
4. Name the label 'Equipment COG'

5. Add the properties Dry CG X, Dry CG Y and Dry CG Z to the label



6. Change the Alias Name of the properties to X, Y and Z as shown below.



7. Click OK to save the label
8. Open Windows Explorer
9. Browse to the SharedContent\Labels\Examples\Equipment COG folder and open the Equipment COG.rfm file
10. Separately, open another Windows Explorer window and browse to \\SharedContent\Labels\Types of Labels\Equipment\Equipment Occurrence Location and open the Equipment Occurrence Location.rfm file

- Copy the content of the <RTF_LABEL> to </RTF_LABEL> in the Equipment Occurrence Location.rfm file and paste it into the Equipment COG.rfm file, overwriting the <RTF_LABEL> to </RTF_LABEL> section in it.

```

Equipment Occurrence Location.rfm - stylesheet03.xsl.xls - Equipment Occurrence Location.rfm
source (xml) xpath console stylesheet(xsl) result result(html)
<?xml version="1.0" encoding="UTF-8"?>
<REPORT_FORMATTING Name="Equipment Occurrence Location" Description="">
  <DESIGN_TIME Progid="SP3DLabelFormatDesigner.RTFLabel" Action="RTFLabel" />
  <RUN_TIME Progid="SP3DLabelsFormat.FormatLabel" Action="RTFLabel" />
  <FORMATTING_PARAMETERS Name="Equipment Occurrence Location" Type="Internal" />
  <RTF_LABEL>
    <POINTS>
      <POINT Name="occurrence Location" X="X" Y="Y" Z="Z" M="M" />
      <READOUT>
        <DIRECTION />
        <VALUE />
      </READOUT>
    </POINTS>
    <BLOCKS>
      <BLOCK Action="Visible">
        <TOKENS>
          <POSITION Visible="Yes" Axis="X" Point="Occurrence Location" />
          <!--FORMAT_EXPRESSION Expression='@{@@@...}'-->
        </TOKENS>
        <TEXT Visible="Yes" Value=" " />
        <POSITION Visible="Yes" Axis="Y" Point="Occurrence Location" />
        <FORMAT_EXPRESSION Expression='@{@@@...}' />
        <POSITION Visible="Yes" Axis="Z" Point="Occurrence Location" />
        <FORMAT_EXPRESSION Expression='@{@@@...}' />
      </BLOCK>
    </BLOCKS>
  </RTF_LABEL>
</REPORT_FORMATTING>

```



```

Equipment COG.rfm - stylesheet01.xsl.xls - Equipment COG.rfm
source (xml) xpath console stylesheet(xsl) result result(html)
<?xml version="1.0" encoding="UTF-8"?>
<REPORT_FORMATTING Name="Equipment COG" Description="">
  <DESIGN_TIME Progid="SP3DLabelFormatDesigner.RTFLabel" Action="RTFLabel" />
  <RUN_TIME Progid="SP3DLabelsFormat.FormatLabel" Action="RTFLabel" />
  <FORMATTING_PARAMETERS Name="Equipment COG" Type="Internal" />
  <RTF_LABEL>
    <POINTS />
    <VECTORS />
    <BLOCKS>
      <BLOCK Action="visible" child="No">
        <TOKENS>
          <TEXT Value="\{\rtf1\ansi\deft0{\>" />
        </TOKENS>
      </BLOCK>
    </BLOCKS>
  </RTF_LABEL>
</REPORT_FORMATTING>

```

- Similarly, open the Equipment COG.rfp file and the Equipment Occurrence Location.rfp file and copy the <UOMS> to </UOMS> and <MATRIXES> to </MATRIXES> sections in the Equipment Occurrence Location.rfp file and paste them into the Equipment COG.rfp file, inserting them above <FIELDS />.

```

Equipment Occurrence Location.rfp - stylesheet07.xsl.xls - Equipment Occurrence Location.rfp
source (xml) xpath console stylesheet(xsl) result result(html)
<?xml version="1.0" encoding="UTF-8"?>
<FORMATTING_PARAMETERS Name="Equipment Occurrence Location" Description="">
  <DESIGN_TIME Progid="" Action="" Arg="" />
  <RUN_TIME Progid="" Action="" Arg="" />
  <UOMS>
    <UOM Name="Location" CanInherit="Yes" Type="DISTANCE" Primary="Yes" />
    <UOM Name="HVOrientationDegMinSec" CanInherit="No" Type="Angle" />
  </UOMS>
  <MATRIXES>
    <MATRIX LinkName="MyTransformA" DisplayName="Equipment Location" />
  </MATRIXES>
</FORMATTING_PARAMETERS>

```

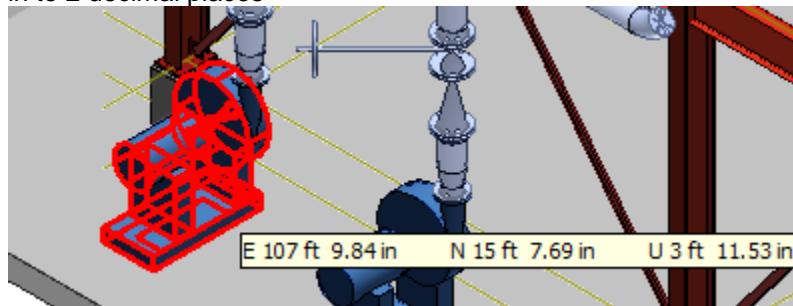


```

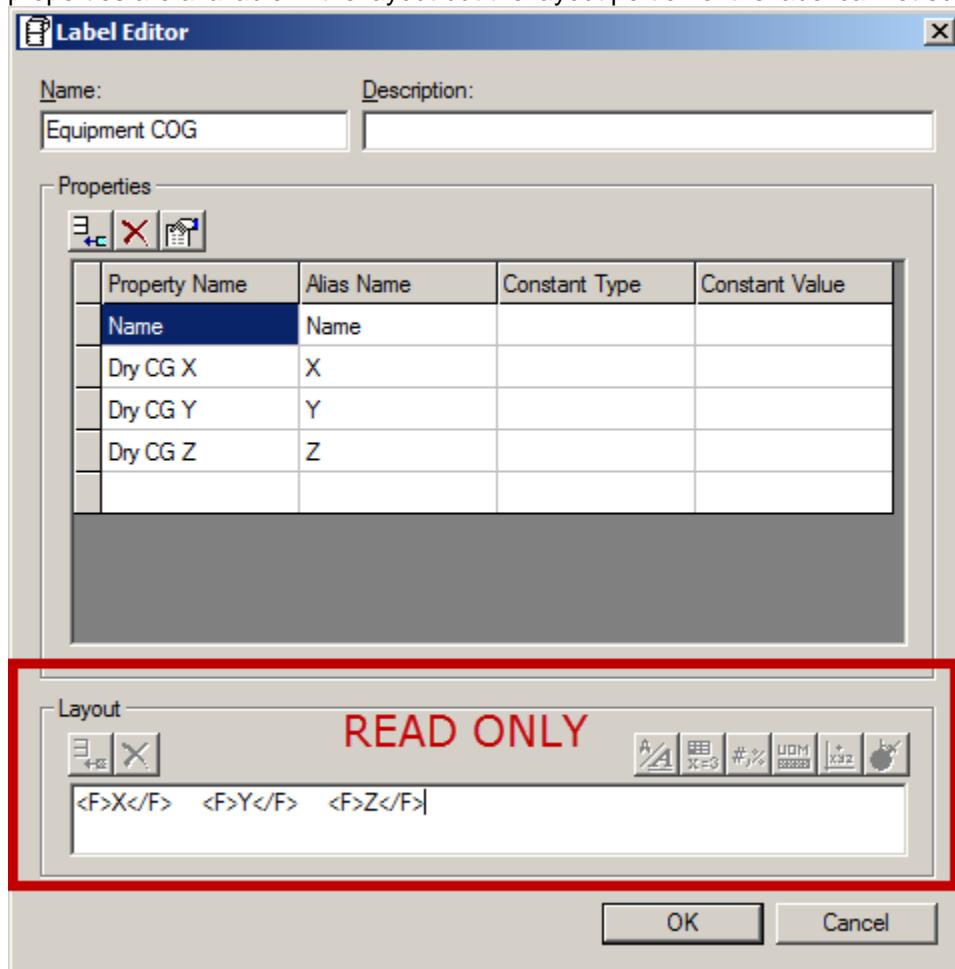
Equipment COG.rfp - stylesheet05.xsl.xls - Equipment COG.rfp :stylesheet05.xls
source (xml) xpath console stylesheet(xsl) result result(html)
<?xml version="1.0" encoding="UTF-8"?>
<FORMATTING_PARAMETERS Name="Equipment COG" Description="">
  <DESIGN_TIME Progid="" Action="" Arg="" />
  <IN_TIME Progid="SP3DPrompts.ct1Tab" Action="" Arg="" />
  <FIELDS />
</FORMATTING_PARAMETERS>

```

- Save all files and exit the XML editor.
- Test the label as a tooltip on Pump-001 in U01. Notice that the coordinates are global and in ft and in to 2 decimal places



15. In the Catalog task, edit the Equipment COG label again. You will see that the X, Y and Z properties are available in the layout but the layout portion of the label cannot be edited.



Editing Units of Measure

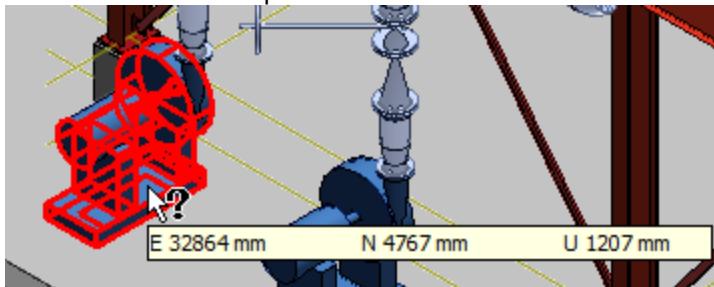
1. Edit the Equipment COG.rfp file
2. Notice the CanInherit flag is set to Yes for the Location UOM. This means that the UOM for DISTANCE is being inherited from the session

```
<UOMS>
  <UOM Name="Location" CanInherit="Yes" Type="DISTANCE" Primary="ft" Secondary="in"
    <UOM Name="HVorientationDegMinSec" CanInherit="No" Type="ANGLE" Primary="deg" Seco
  </UOMS>
```

3. To fix the UOM to mm, set CanInherit =No and Primary = mm and secondary = ""

```
<UOM
  Name="Location"
  CanInherit="No"
  Type="DISTANCE"
  Primary="mm"
  Secondary=""
  Tertiary=""
  UnitsDisplayed="yes"
  PrecisionType="Decimal"
  DecimalPrecision="0"
  FractionalPrecision=""
  Leadingzero=""
  Trailingzeros=""
  ReduceFraction="Yes" />
```

4. Exit the Smart 3D session (save session when prompted) and re-enter it.
 5. Test the label as tooltip to see the effects



Editing the Coordinate System

1. Edit the Equipment COG.rfp file
2. Notice the CanInherit flag is set to Yes for the Matrix MyTransformA. This means that the coordinate system w.r.t which the points are shown is being inherited from the session (in the case of tooltip, from the pinpoint toolbar)
3. To fix the coordinate system to a known coordinate system, enter its name in the ModelName field and set CanInherit = "No". For example to show coordinates w.r.t U01 CS, set as below

```
<MATRIXES>
  <MATRIX
    LinkName="MyTransformA"
    DisplayName="Equipment Location"
    ModelName="U01 CS"
    CSType="0"
    PosX="E "
    PosY="N "
    PosZ="U "
    NegX="W "
    NegY="S "
    NegZ="D "
    XYreferenceAxis="1"
    CanInherit="No" />
</MATRIXES>
```

Note that this coordinate system set via ModelName is not being honored for tooltips and embedded labels. It is honored by drawing labels. A software fix is needed to honor it for all users of the label.

COM Queries

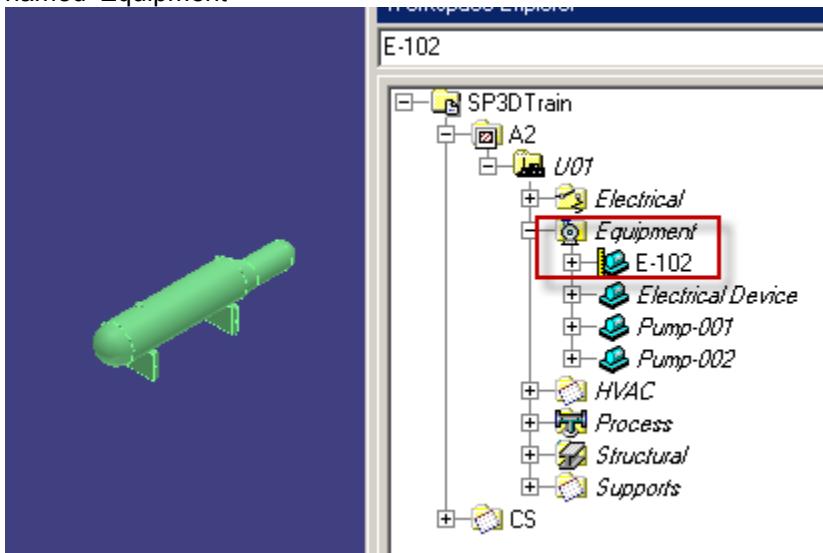
Define a label query to report the parent system of equipment items

Objective: After this lab, you will be able to

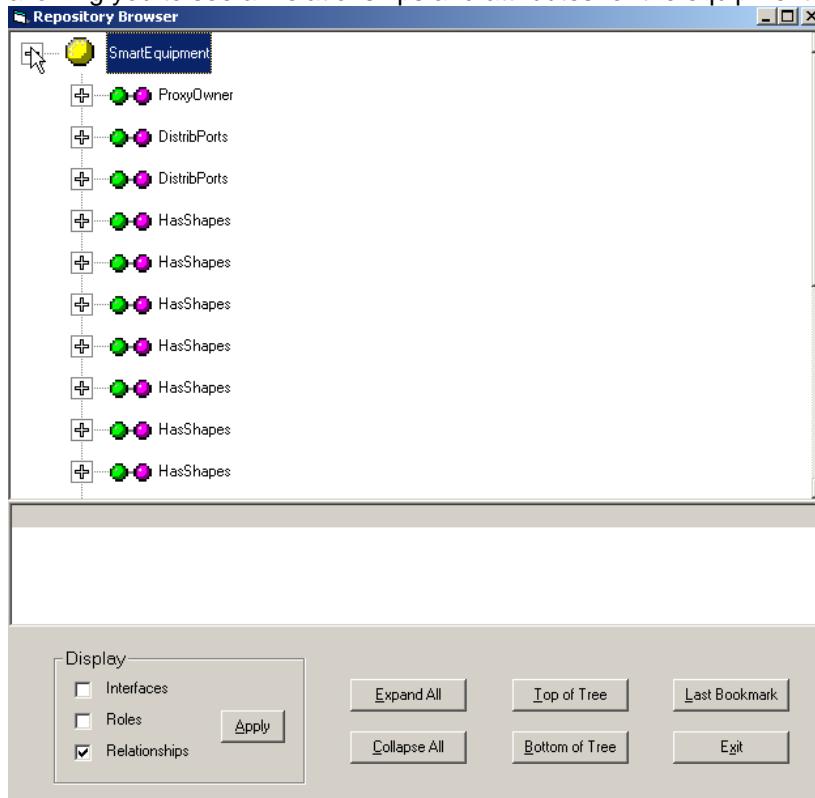
- Trace relationships using repository browser
- Define a COM label query for equipment to system relationship

Tracing relationship in repository browser

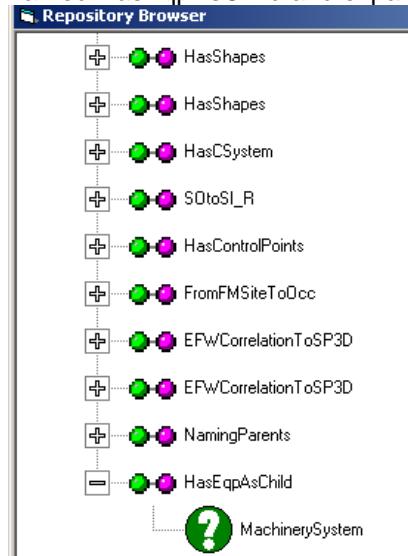
1. Open Smart 3D and define a workspace using Plant Filters\Training Filters\U01 filter
2. Select the equipment E-102. For workspace explorer you can see that it belongs to the system named 'Equipment'



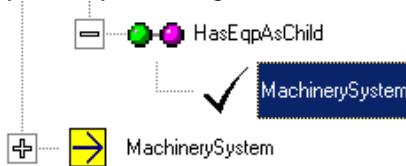
3. Select the equipment E-102 and press Ctrl – Shift – R. This opens the repository browser allowing you to see all relationships and attributes for the equipment and related objects.



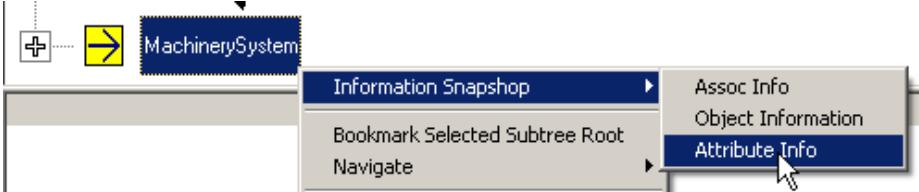
4. From the names of the relationships, we can usually get a idea about what kind of object the relationship is to. E.g. DistribPorts is a relationship to port, HasShapes is a relationship to shapes etc
 5. Typically parent relationships are at the bottom of the list. Scroll down till you find relationship named HasEqpAsChild and expand it by clicking + sign



6. This tells us that there is a relationship named HasEqpAsChild between our select SmartEquipment and a MachinerySystem. Let us verify that we have the correct machinery system by checking its name. Click on MachinerySystem and it opens a new yellow node.



7. Select the MachinerySystem node, then right click on the new yellow MachinerySystem node and select AttributeInfo



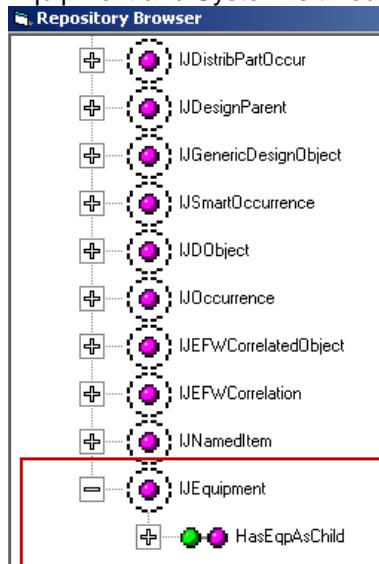
8. Scroll in the bottom pane using the arrow keys till you see the IJNamedItem interface and Name attribute that verifies the name

Item	Object Name	Interface Name	Interface ID	Attribute Name	Value	Type	Unit Type
Object With Attri...	MachinerySystem	IJNamedItem	(7330A824-789...	ApprovalStatus	1	Long	None
				ApprovalReason	-1	Long	None
Object With Attri...	MachinerySystem	IJDesignParent	(524AAFA0-1E...	Name	Equipment	String	None
				IIDForTypeString	Equipment	String	None

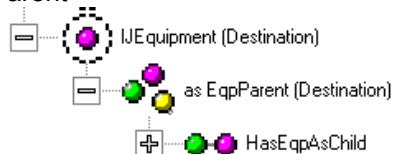
9. Check the 'Interfaces' box under 'Display' and click 'Apply'



10. Expand ‘SmartEquipment’ and scroll to the bottom. You will see that relationship between Equipment and System is through IJEquipment interface.



11. Select ‘Roles’ box and click ‘Apply’
12. Expand ‘SmartEquipment’, scroll to the bottom and see the role on equipment side is called ‘EqpParent’

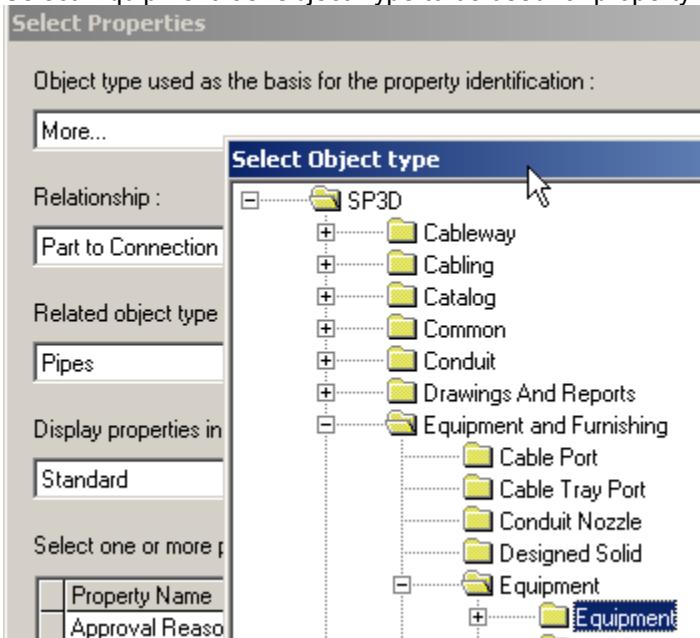


Defining a COM Label

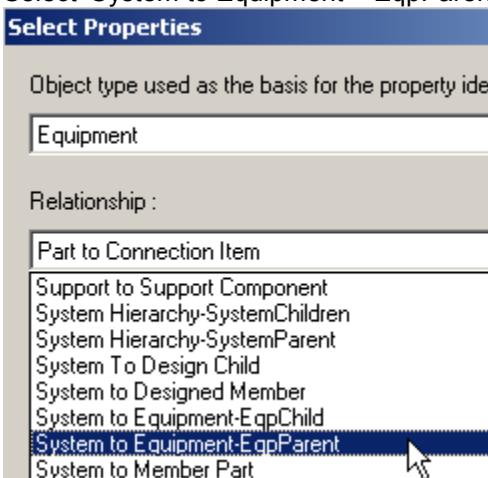
13. In the catalog task, create a new COM label named ‘Equipment Parent COM’
14. Remove existing property ‘Name’ and click ‘Add...’ button



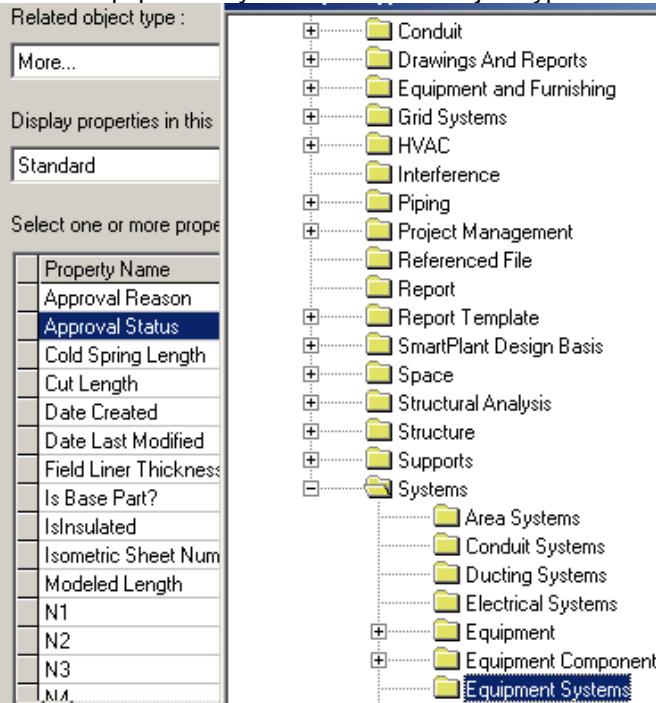
15. Select 'Equipment' as 'Object Type to be used for property identification'



16. Select 'System to Equipment – EqpParent' as the relationship



17. Select 'Equipment System' as related object type



18. Select 'Name' as the property and click OK.

19. Insert the name property to Layout and click OK.

20. Now open Windows explorer and find the Labels\Examples\Equipment Parent COM folder on your symbols share.

21. Open the Equipment Parent COM.rqe file. We see the interface and relationcollection (role) name

```
<RETURNED_PROPERTY
  Name="Name"
  SQLType="BStr">
  <PATHS>
    <PATH
      SourceType="IJEquipment"
      SourceBOC="SmartEquipment"
      DestinationInterface="IJNamedItem"
      DestinationProperty="Name"
      DestinationBOC="EquipmentSystems"
      Concatenate="No"
      PathSeparator="">
      <STROKES>
        <STROKE
          Interface="IJEquipment"
          RelationCollection="EqpParent"
          Recursive="No"
          Filter="First"
          IsVirtualRelationship="No" />
      </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>
```

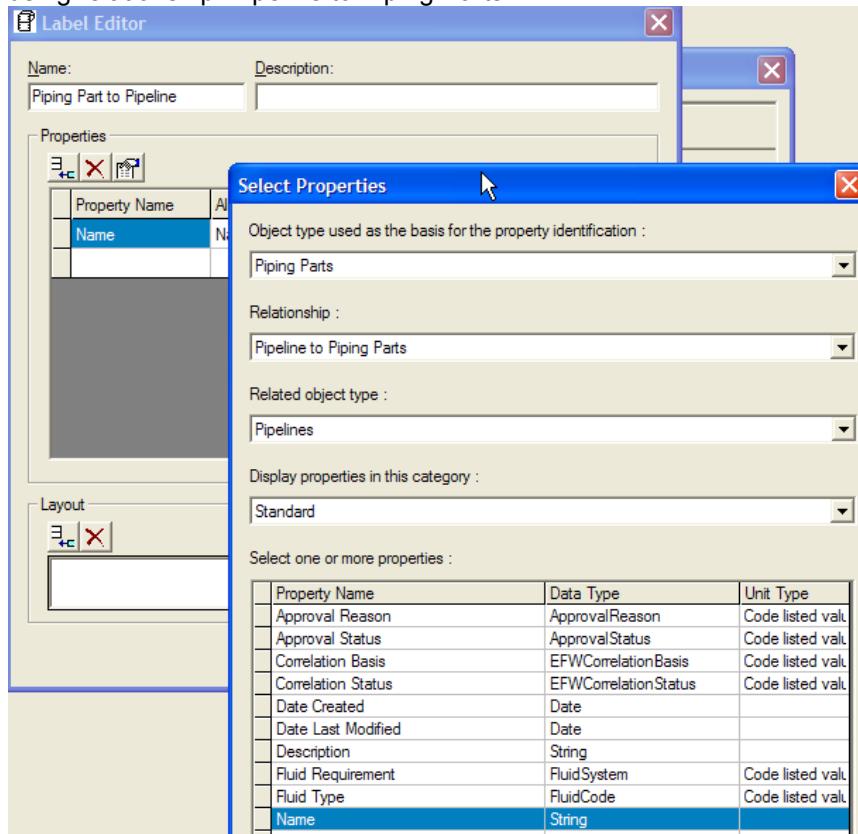
22. Test the label in equipment task as a tooltip.

Combining Labels by editing RQE files

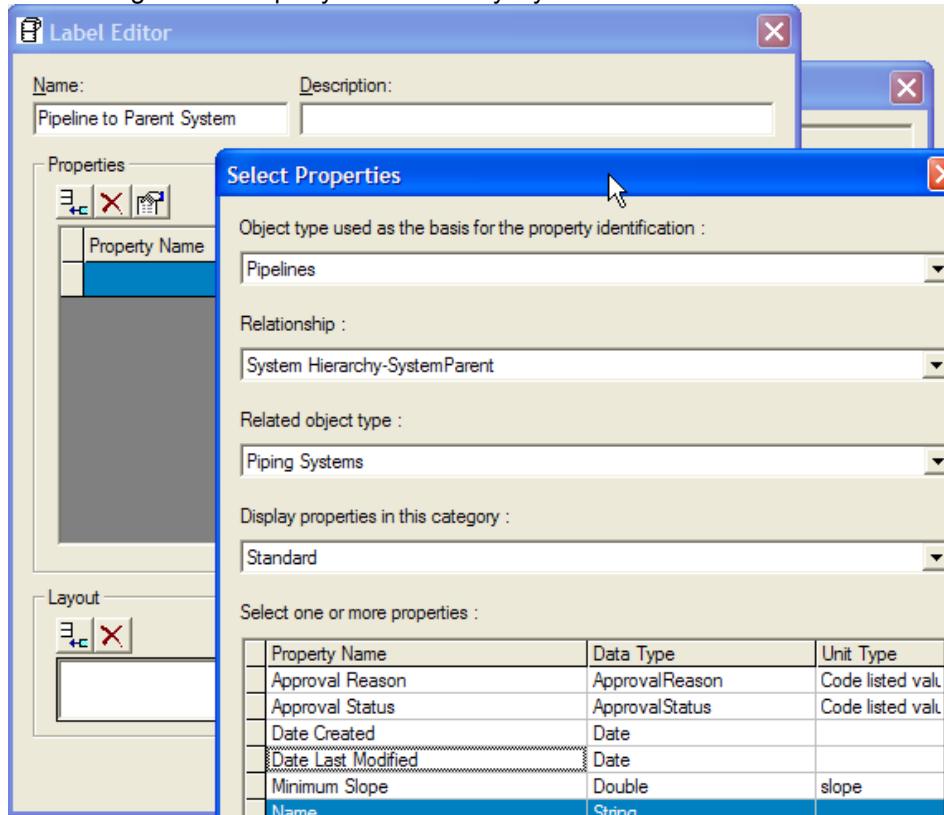
Objective: After completing this lab, you will be able to

- Combine queries from two COM labels into a single label

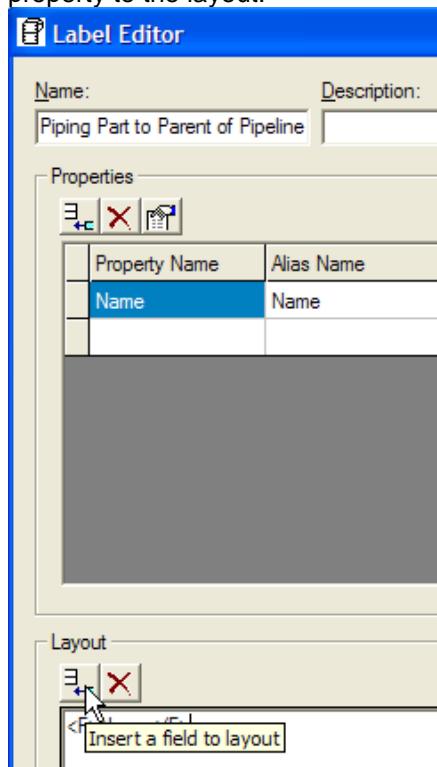
1. Define a label 'Piping Part to Pipeline' that starts at piping parts and returns pipeline name using relationship 'Pipeline to Piping Parts'



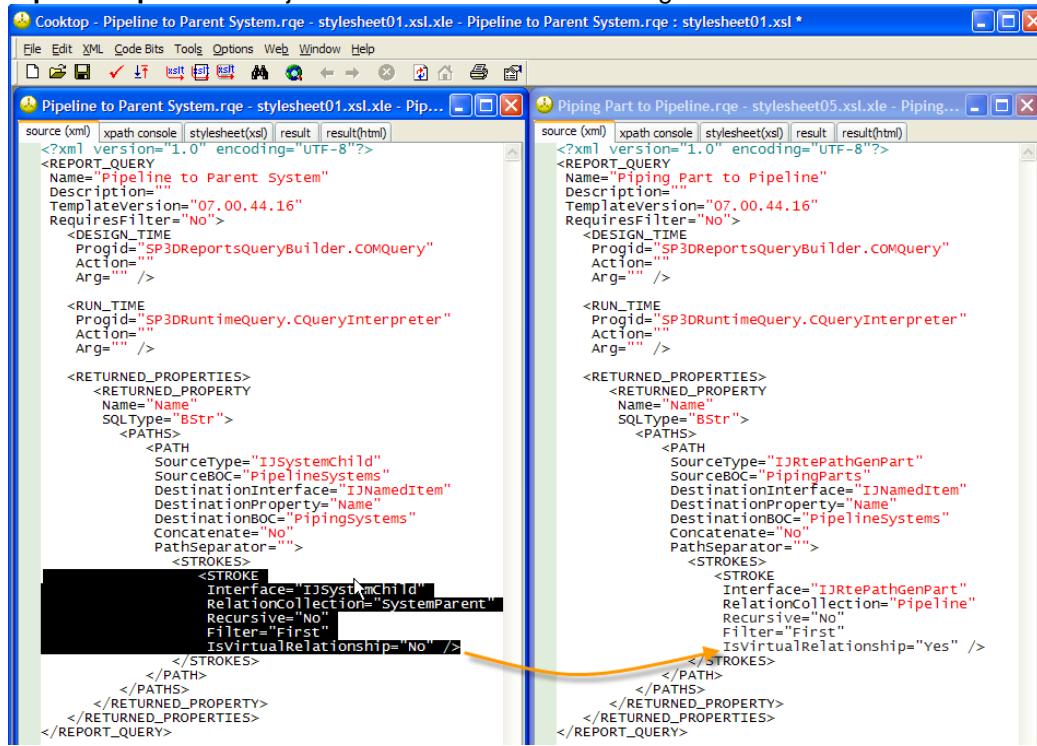
2. Define a label 'Pipeline to Parent System' that starts at pipelines and returns piping system name using relationship 'SystemHierarchy-SystemParent'



3. Define a label named 'Piping Part to Parent of Pipeline' that just adds the default 'Name' property to the layout.

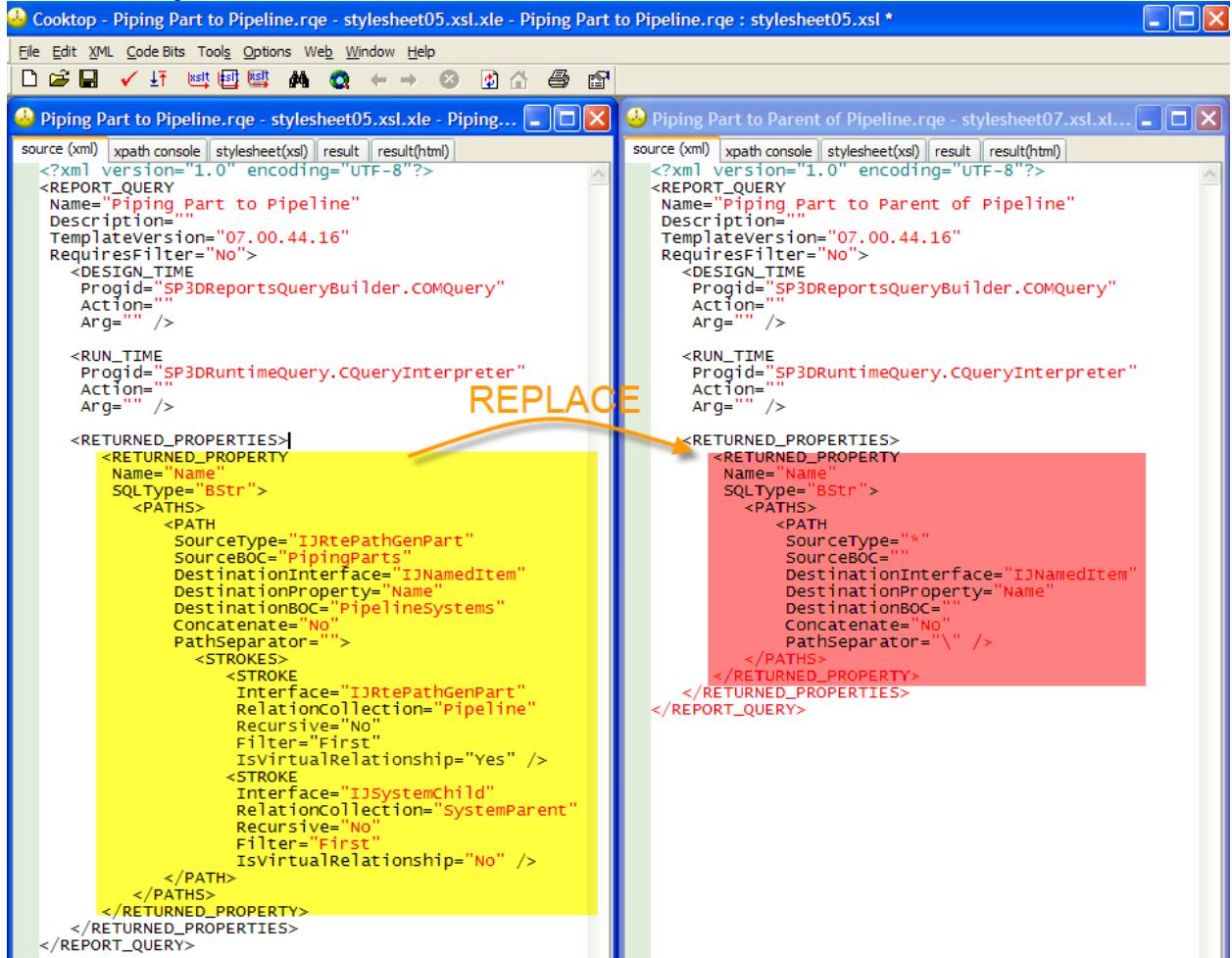


4. Open all rqe files **Piping Part to Pipeline.rqe** and **Pipeline to Parent System.rqe** in CookTop XML Editor.
5. Copy the STROKE from the **Pipeline to Parent System.rqe** to the **Piping Part to Pipeline.rqe** as shown just before the </STROKES> tag.



6. Open the **Piping Part to Parent of Pipeline.rqe** file in CookTop.

7. Copy <RETURNED_PROPERTY> to </RETURNED_PROPERTY> in the **Piping Part to Pipeline.rqe** file and paste it in the **Piping Part to Parent of Pipeline.rqe**, replacing the section existing there.



8. Validate the file using Validate command (keyboard shortcut F7)
9. Save all files and exit CookTop.
10. Test the label by using it as a tooltip for Piping Parts, you should see the name of the parent of the pipeline. e.g. in A2 – U01 you should see ‘Process’, in A2 – U03, you should see ‘Water’ and in Area01 – Unit01 you should see ‘Unit01’

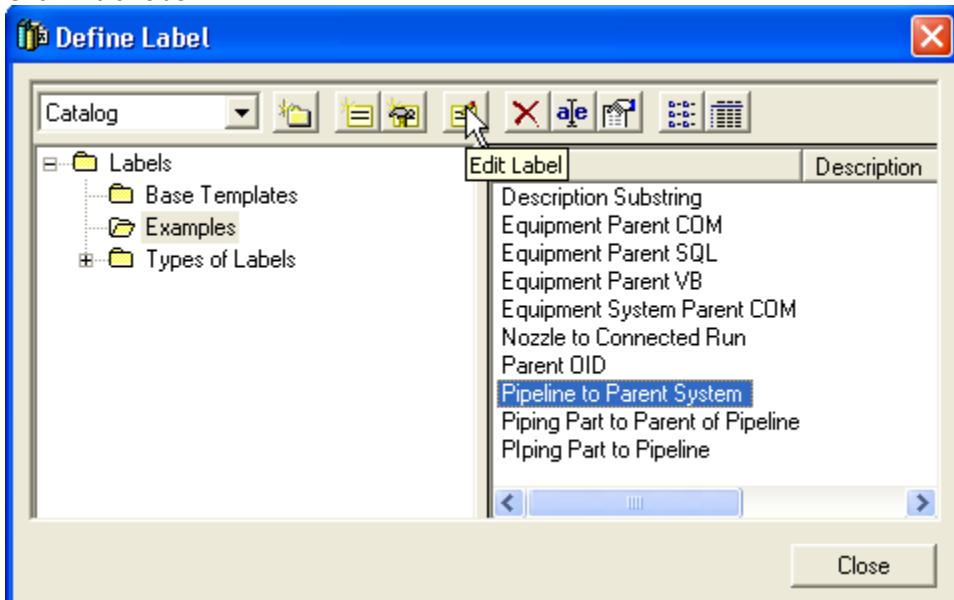
Recursive querying in labels

Objective: After completing this lab, you will be able to

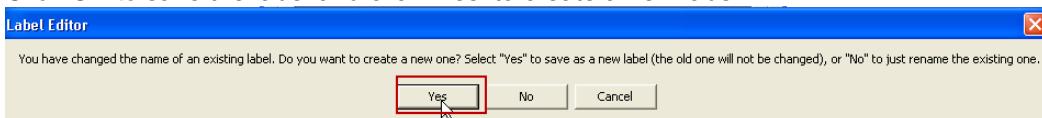
- Follow a relationship recursively repeatedly for a set number of times. This is a shortcut for repeating a STROKE in the query.

1. Start SmartPlant 3D and define a workspace with filter Plant Filters\Training Fitlers\U01
2. Switch to Catalog Task
3. Tools → Define Label and select the Examples\‘Pipeline to Parent System’ label

4. Click 'Edit Label'



5. Rename the label to 'Pipeline to All Parents'
6. Click OK to save the label and click 'Yes' to create a new label.



7. Accept the location offered 'Labels\Examples' and click OK.
8. In Windows Explorer navigate to 'SharedContent\Labels\Examples\Pipeline to All Parents'
9. Open the Pipeline to All Parents.rqe file
10. Change 'Recursive' to 'Yes' and Concatenate to 'Yes'

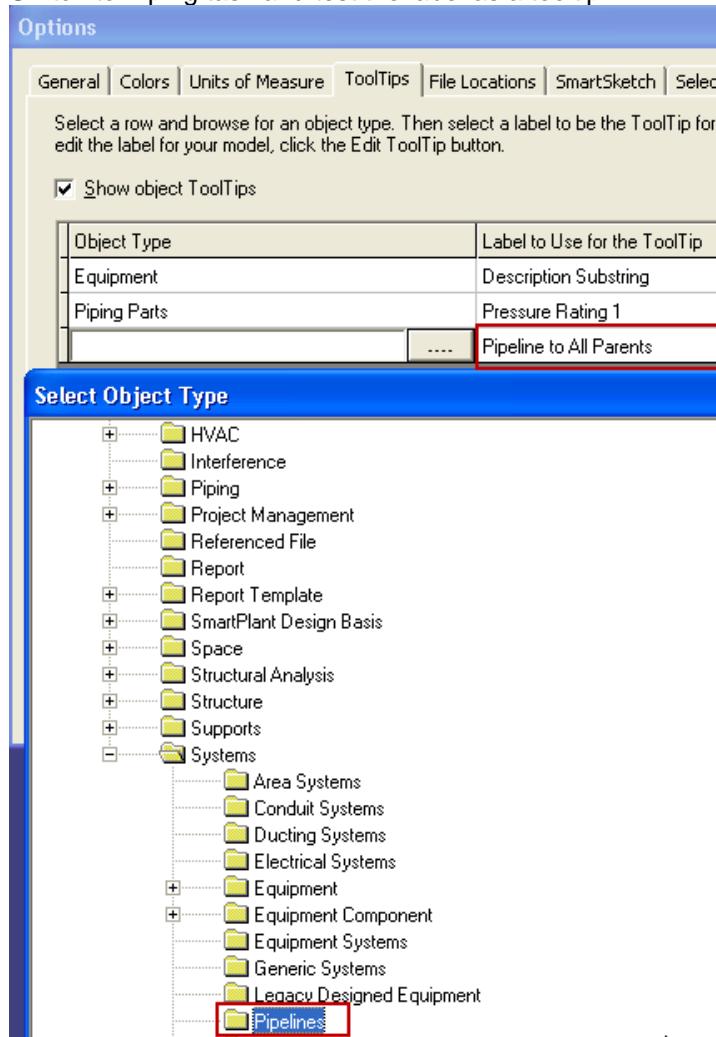
```

<RETURNED_PROPERTY
  Name="Name"
  SQLType="BStr">
  <PATHS>
    <PATH
      SourceType="IJSystemChild"
      SourceBOC="Pipelinesystems"
      DestinationInterface="IJNamedItem"
      DestinationProperty="Name"
      DestinationBOC="Pipingsystems"
      Concatenate="Yes"
      PathSeparator="\">
        <STROKES>
          <STROKE
            Interface="IJSystemChild"
            RelationCollection="SystemParent"
            Recursive="Yes"
            Filter="First"
            IsVirtualRelationship="No"
            ExitValue="1" />
        </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>

```

11. Save the file and exit editor.

12. Switch to Piping task and test the label as a tooltip.



Recursive label using “Implements” ExitCondition

Objective: After completing this lab, you will be able to

- Follow a relationship recursively repeatedly until an ExitCondition “Implements” is met, meaning an object which implements a particular interface (which is an object of a particular type) is found

- Switch to Catalog task and select the ‘Pipeline to All Parents’ label
- Edit label and rename the label to ‘Pipeline to Area Parent’
- Click OK to save the label.
- Click Yes to create a new label.
- Click OK to accept the location.
- In Windows Explorer navigate to ‘SharedContent\Labels\Examples\Pipeline to Area Parent’
- Open the Pipeline to Area Parent.rqe file.

8. Change Concatenate to 'No', ExitValue to "IJAreaSystem" and add the line ExitCondition = "Implements"

```
<RETURNED_PROPERTY
  Name="Name"
  SQLType="Bstr">
  <PATHS>
    <PATH>
      SourceType="IJSystemChild"
      SourceBOC="PipelineSystems"
      DestinationInterface="IJNamedItem"
      DestinationProperty="Name"
      DestinationBOC="PipingSystems"
      Concatenate="No"
      PathSeparator="\"/>
      <STROKES>
        <STROKE
          Interface="IJSystemchild"
          RelationCollection="SystemParent"
          Recursive="Yes"
          Filter="First"
          IsVirtualRelationship="No"
          Exitcondition="Implements"
          Exitvalue="IJAreaSystem" />
      </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>
```

9. Save the file and exit editor.
10. Switch to Piping task and test the new label as tooltip.

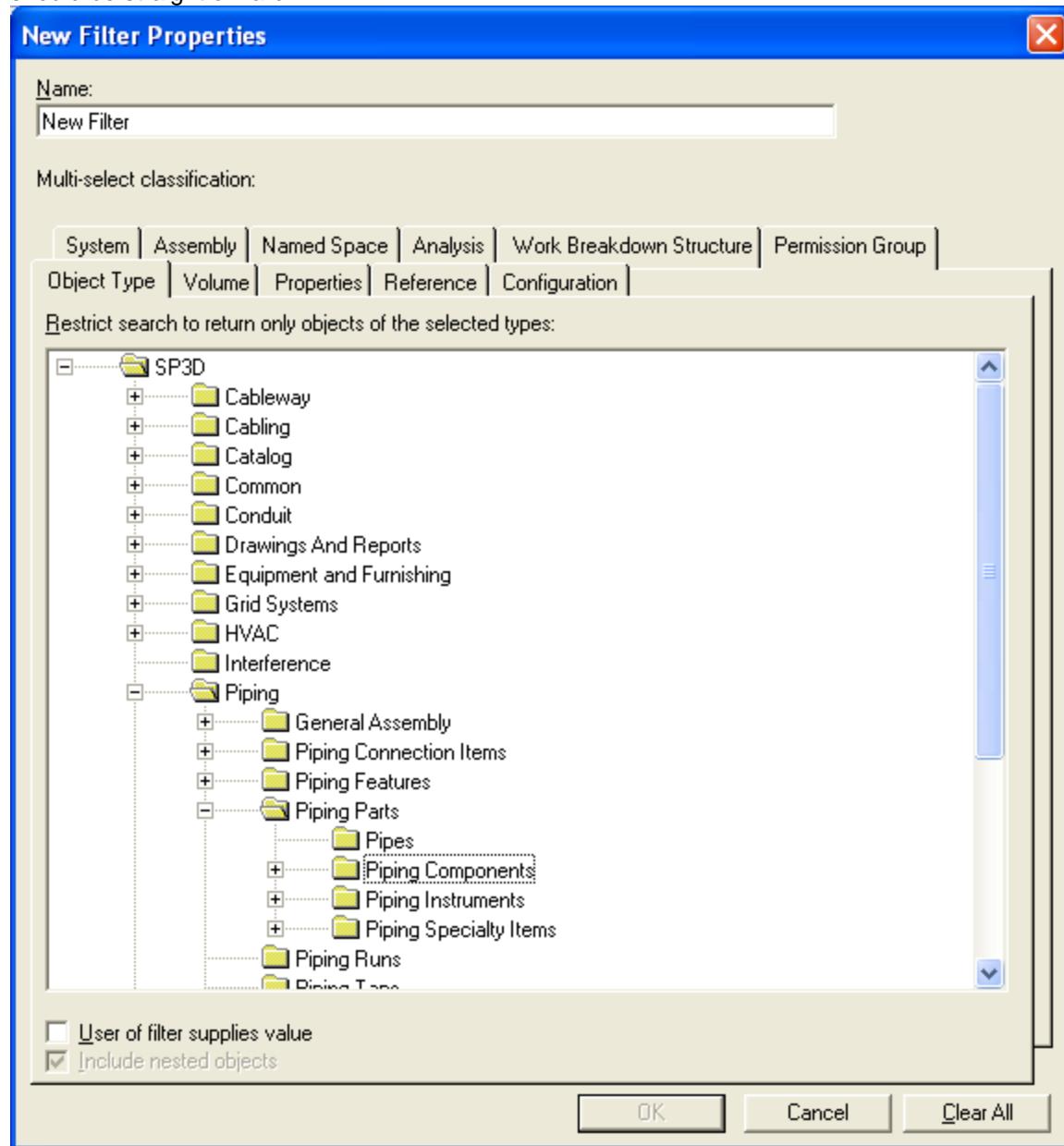
Multi-path label query

Objective: After completing this lab, you will be able to

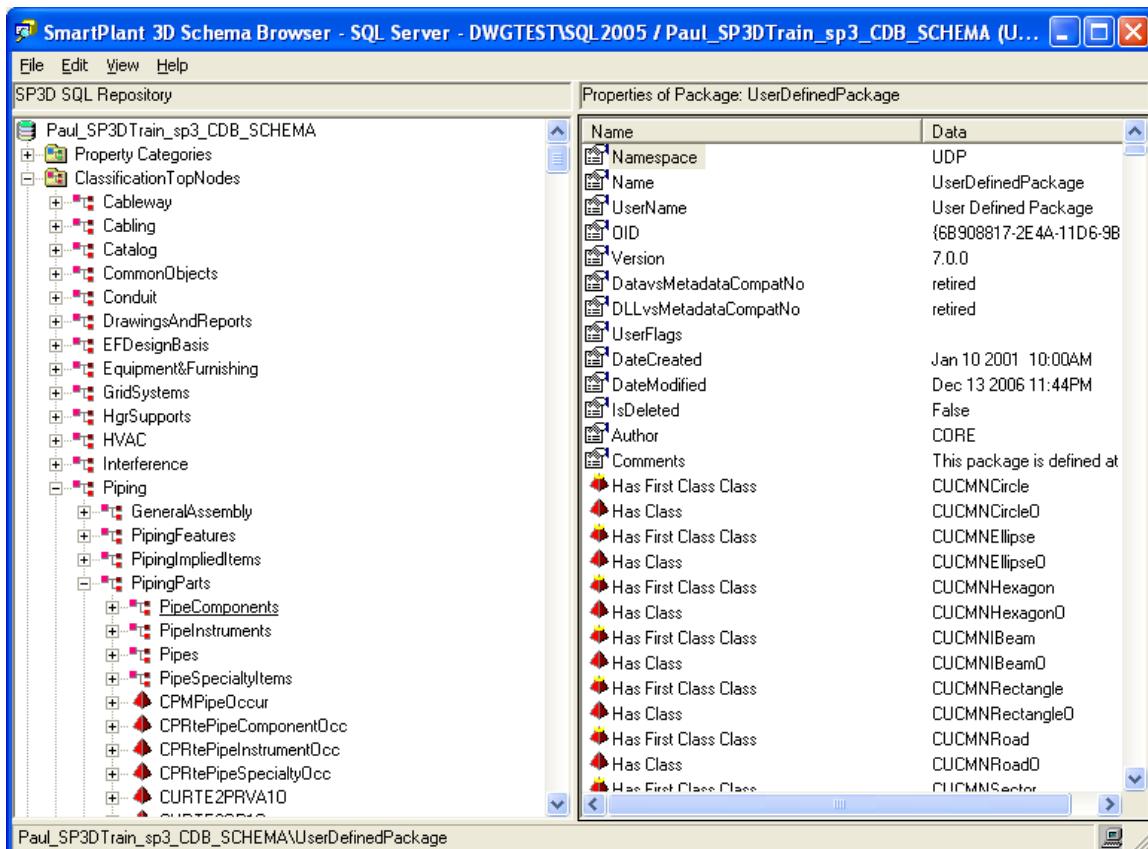
- Conditionally execute different queries in the same label, based on the type of object being evaluated in the query.

1. Switch to Catalog task.
2. Tools → Define Label
3. Create a new COM label in Labels\Examples folder
4. Name the label 'Component Type'
5. Select the 'Name' property and Insert it to layout.
6. Click OK to save the label.
7. In Windows Explorer navigate to 'Symbols\Labels\Examples\Component Type' folder
8. Open the Component Type.rqe file.
9. Since the objects that we'll be labeling with this label can be potentially of different kinds, we have to choose only the ones of Piping Component type. We do this by finding an Interface that only Piping Components implement and then plugging its name in the SourceType attribute. To do that, we use the *Classification Top Nodes* part of the *Schema Browser* tool.

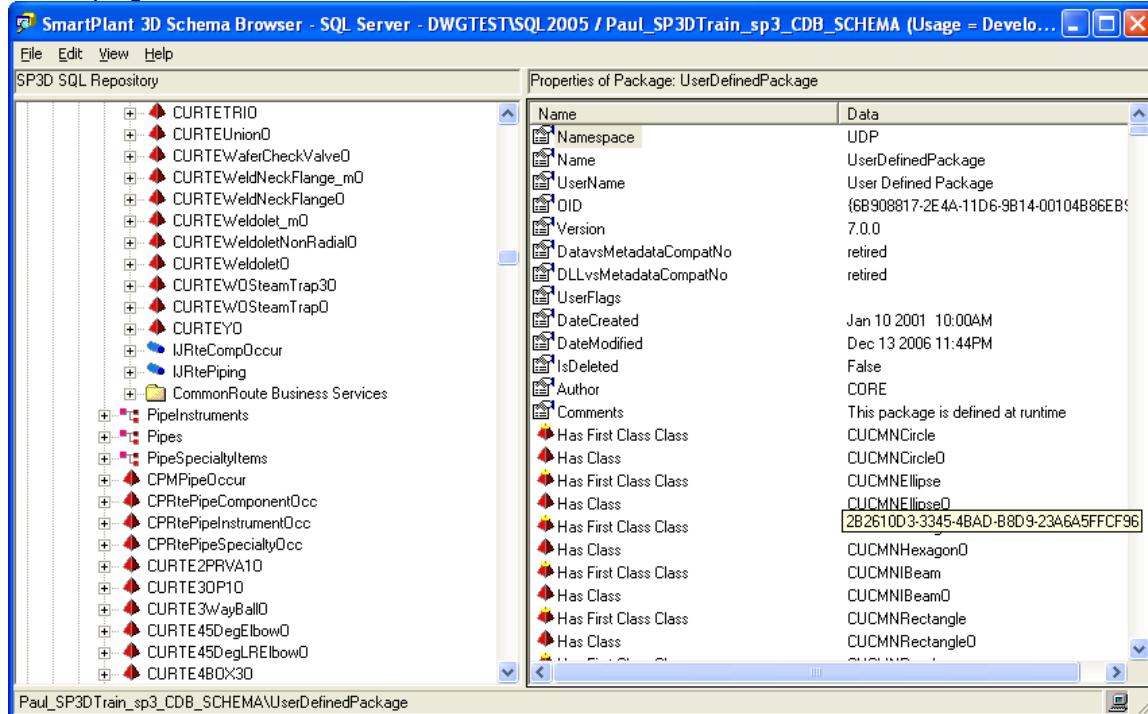
10. Let's find in the *Schema Browser Classification Top Nodes* the node corresponding to the *Piping Components* node in the SP3D Object Type Hierarchy. Since those are the same hierarchies, this should be straightforward



6.



11. Now let's expand the *PipeComponents* node in the *Schema Browser*. If we scroll down to the interfaces that are implemented by that node we'll see 2 of them: *IJRteCompOccur* and *IJRtePiping*.



Notes:

- The interfaces listed under a given node in the *Classification TopNodes Hierarchy* are not all the interfaces implemented by that object type but rather the “type defining” interfaces – i.e. for an object to be classified as a Piping Component it HAS to implement those interfaces.
 - We need only one interface to uniquely identify an object’s type, but since the interfaces listed under a given node are also used to determine the object type’s place in the overall *Business Object Classification* (BOC) hierarchy, we could have 2 or more as is our case.
 - It is usually very easy and intuitive to pick the defining one, though: in our case *IJRtePiping* seems too general, and indeed, a quick check in few of the other nodes shows that all piping objects are required to implement it. So, this leaves us with *IJRteCompOccur* as the interface that uniquely identifies objects of type Piping Component.
12. Now for the part where we need to apply the predefined text description “COMP”. We achieve this by writing “CONSTANT” for DestinationInterface, and then the constant value/description “COMP” in DestinationProperty.

```
<RETURNED_PROPERTY
  Name="Name"
  SQLType="Bstr">
  <PATHS>
    <PATH
      SourceType="IJRteCompOccur"
      SourceBOC=""
      DestinationInterface="CONSTANT"
      DestinationProperty="COMP"
      DestinationBOC=""
      Concatenate="No"
      PathSeparator="\\" />
    </PATHS>
  </RETURNED_PROPERTY>
```

Notes:

- To make the label work for any one of those, we just put their respective paths between the <PATHS> </PATHS> nodes.
- Also, note that those different paths are all under a single <RETURNED_PROPERTY> node. Given an object, the software will try to execute the instructions in the first <PATH> node under each property (<RETURNED_PROPERTY> node) when trying to extract the value of that property, if the object is not of the SourceType of the first <PATH> it will try the second. If the second one fails too, it will then try the 3rd and so on. (This is also true if the execution of the PATH statement fails for any other reason.)

13. Pipe Parts

```
<PATH
  SourceType="IJRtePipeOccur"
  SourceBOC=""
  DestinationInterface="CONSTANT"
  DestinationProperty="PIPE"
  DestinationBOC=""
  Concatenate="No"
  PathSeparator="\\" />
```

14. Piping Instruments

```
<PATH  
  SourceType="IJRteInstrumentoccur"  
  SourceBOC=""  
  DestinationInterface="CONSTANT"  
  DestinationProperty="INST"  
  DestinationBOC=""  
  Concatenate="No"  
  PathSeparator="\\" />
```

15. Piping Specialty Items

```
<PATH  
  SourceType="IJRtespecialtyoccur"  
  SourceBOC=""  
  DestinationInterface="CONSTANT"  
  DestinationProperty="SPCL"  
  DestinationBOC=""  
  Concatenate="No"  
  PathSeparator="\\" />
```

16. Support Assemblies

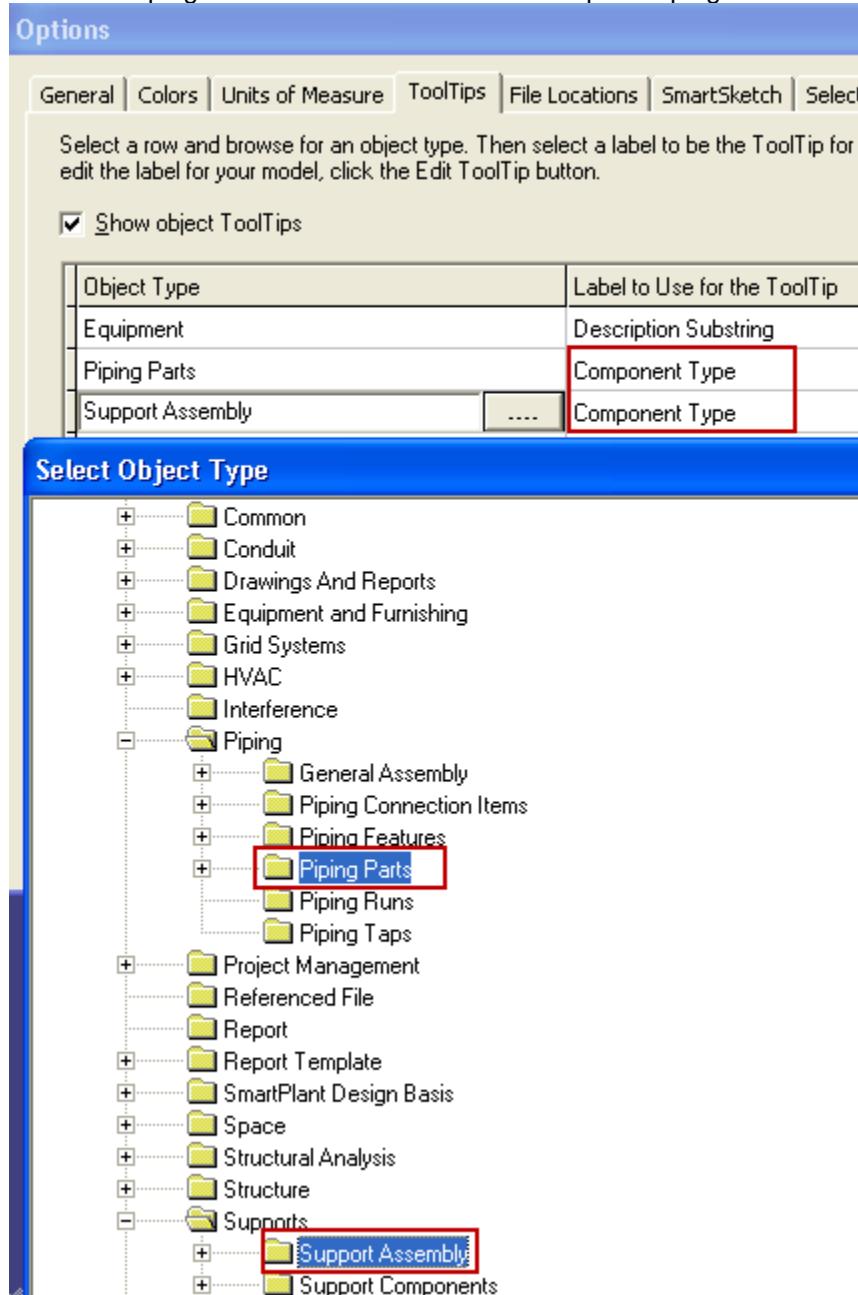
```
<PATH  
  SourceType="IJHgrSupportAssemblyItem"  
  SourceBOC=""  
  DestinationInterface="CONSTANT"  
  DestinationProperty="SUPP"  
  DestinationBOC=""  
  Concatenate="No"  
  PathSeparator="\\" />
```

17. Then the entire label RETURNED PROPERTY looks like

```
<RETURNED_PROPERTY  
  Name="Name"  
  SQLType="Bstr">  
  <PATHS>  
    <PATH  
      SourceType="IJRteCompOccur"  
      SourceBOC=""  
      DestinationInterface="CONSTANT"  
      DestinationProperty="COMP"  
      DestinationBOC=""  
      Concatenate="No"  
      PathSeparator="\\" />  
  
    <PATH  
      SourceType="IJRtePipeOccur"  
      SourceBOC=""  
      DestinationInterface="CONSTANT"  
      DestinationProperty="PIPE"  
      DestinationBOC=""  
      Concatenate="No"  
      PathSeparator="\\" />  
  
    <PATH  
      SourceType="IJRteInstrumentOccur"  
      SourceBOC=""  
      DestinationInterface="CONSTANT"  
      DestinationProperty="INST"  
      DestinationBOC=""  
      Concatenate="No"  
      PathSeparator="\\" />  
  
    <PATH  
      SourceType="IJRteSpecialtyOccur"  
      SourceBOC=""  
      DestinationInterface="CONSTANT"  
      DestinationProperty="SPCL"  
      DestinationBOC=""  
      Concatenate="No"  
      PathSeparator="\\" />  
  
    <PATH  
      SourceType="IJHgrSupportAssemblyItem"  
      SourceBOC=""  
      DestinationInterface="CONSTANT"  
      DestinationProperty="SUPP"  
      DestinationBOC=""  
      Concatenate="No"  
      PathSeparator="\\" />  
  </PATHS>  
</RETURNED_PROPERTY>
```

18. Save the file and exit editor.

19. Switch to Piping task and test the label as a tooltip for Piping Parts and Support Assembly types.



Multi-stroke relationship path in query and using edge

Objective: After you complete this lab, you will be able to

- Define a label that returns connected pipe run name for equipment nozzle using multiple strokes
- Define a label that returns connected pipe run name for equipment nozzle using an edge

Label using multiple strokes

1. Switch to Catalog task

2. Tools → Define Label
3. Select the Examples folder and click 'New COM Label'
4. Name the label 'Nozzle to Connected Run – No Edge'
5. Click OK to save the label.
6. Navigate to folder on symbol share and open the rqe file.
7. Replace <PATH> to </> with <PATH>-</PATH> from the supplied NozzleToRun.txt file

```
<PATH
  SourceType="IJDistribPort"
  DestinationInterface="IJNamedItem"
  DestinationProperty="Name"
  Concatenate="No"
  PathSeparator="\"
  <STROKES>
    <STROKE
      Interface="IJLogicalDistPort"
      RelationCollection="LogDistConn"
      Recursive="No"
      Filter="First"
      IsVirtualRelationship="No"
      ExitValue="1" />

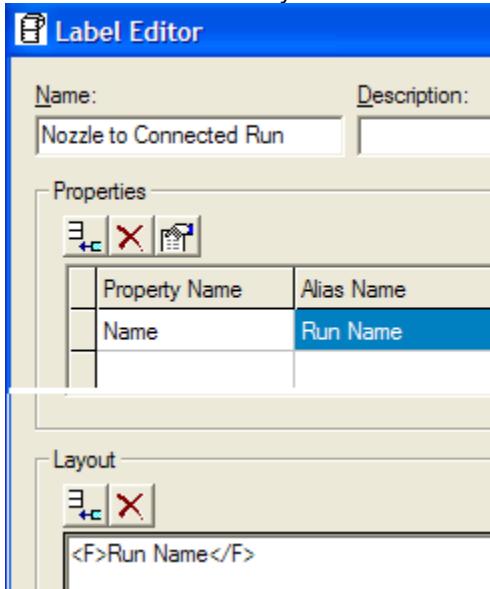
    <STROKE
      Interface="IJDesignParent"
      RelationCollection="DistribConnection"
      Recursive="No"
      Filter="First"
      IsVirtualRelationship="No"
      ExitValue="1" />

    <STROKE
      Interface="IJDistribConnection"
      RelationCollection="Parts"
      Recursive="No"
      Filter="First"
      IsVirtualRelationship="No"
      ExitValue="1" />

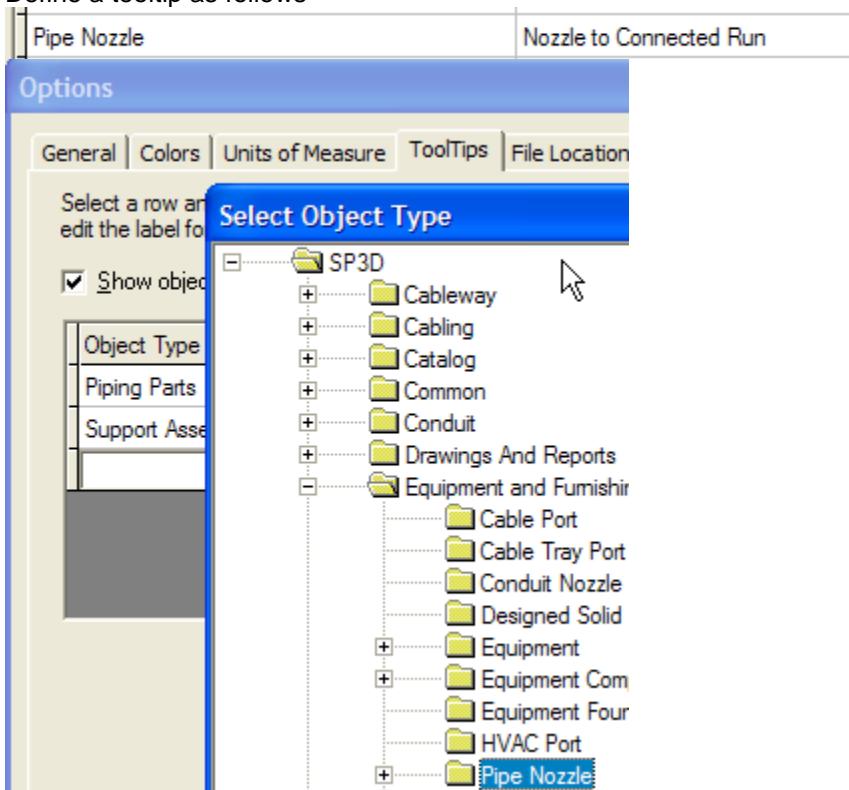
    <STROKE
      Interface="IJRtePathGenPart"
      RelationCollection="Owner"
      Recursive="No"
      Filter="First"
      IsVirtualRelationship="No"
      ExitValue="1" />
  </STROKES>
</PATH>
```

8. Save the rqe file
9. Edit the label in catalog task.
10. Change the alias name to 'Run Name'

11. Click 'Insert a field to layout' and add it to the layout.



12. Click OK to save the label.
13. Switch to Equipment task
14. Define a tooltip as follows

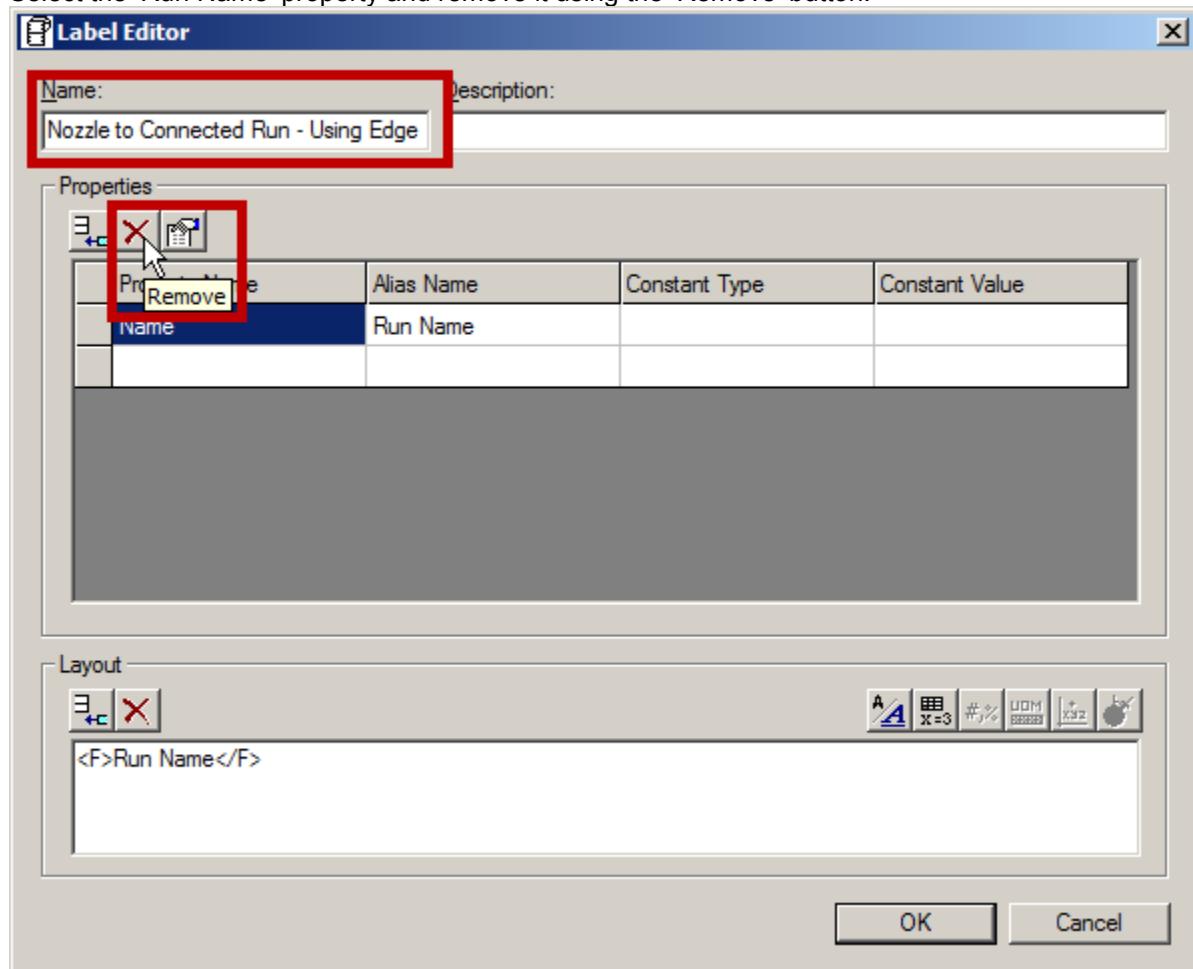


15. Click OK, hover over a nozzle that is connected to a pipe run e.g. the pump discharge nozzles

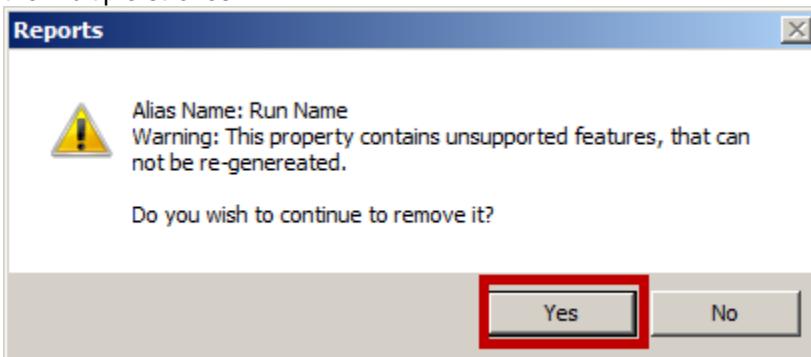
in U01

Label using Edge

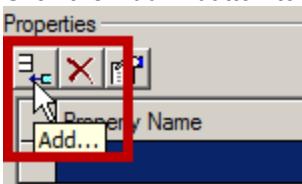
1. Switch to Catalog task
2. Tools → Define Label
3. Select the label 'Nozzle to Connected Run – No Edge' and Edit Label.
4. Change the name of the label to 'Nozzle to Connected Run – Using Edge'
5. Select the 'Run Name' property and remove it using the 'Remove' button.



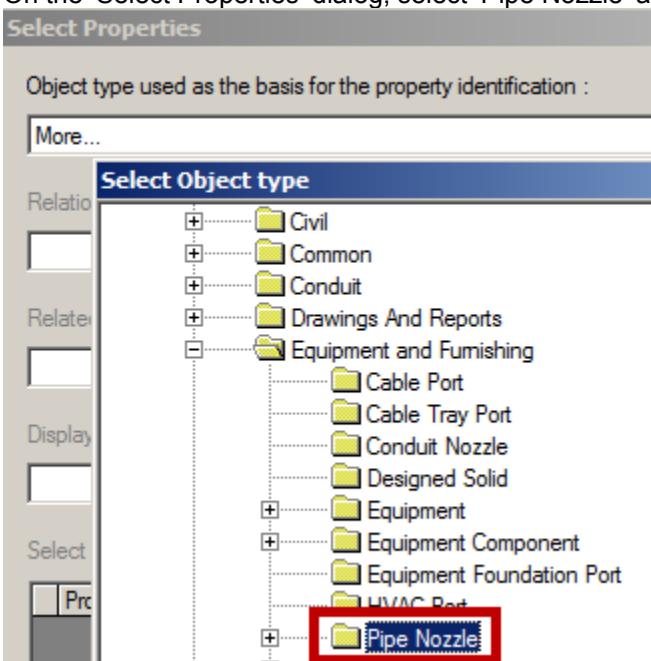
6. A warning message is shown, click **Yes** to the warning. The unsupported features in this case are the multiple strokes.



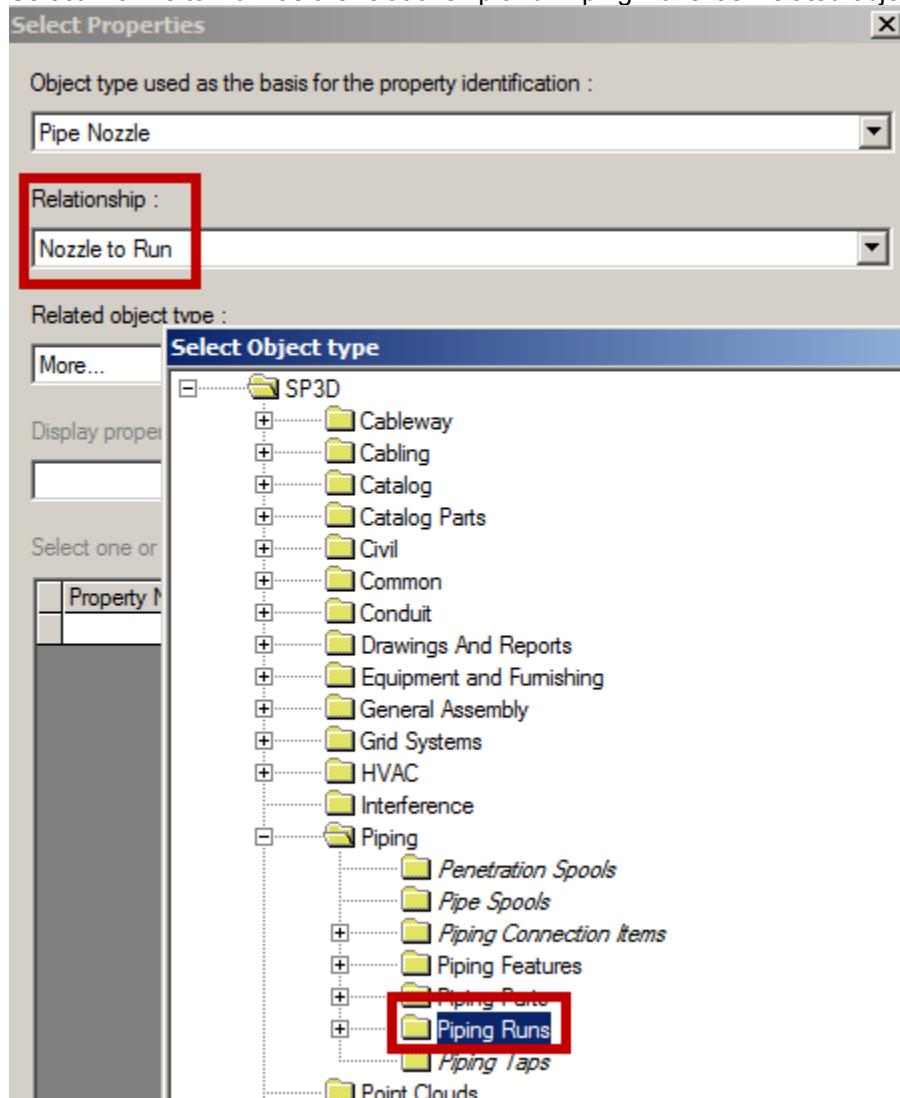
7. Click the Add... button to add a property



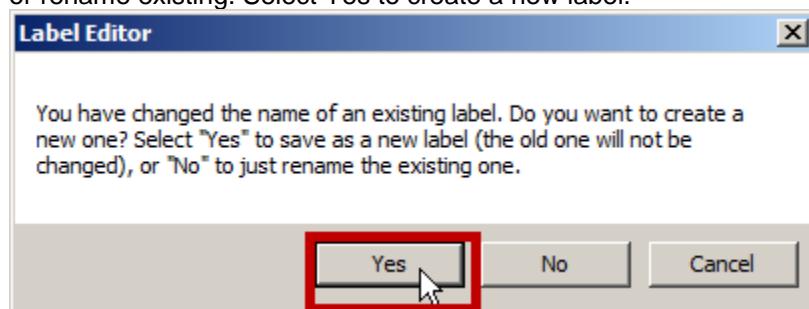
8. On the 'Select Properties' dialog, select 'Pipe Nozzle' as the object.



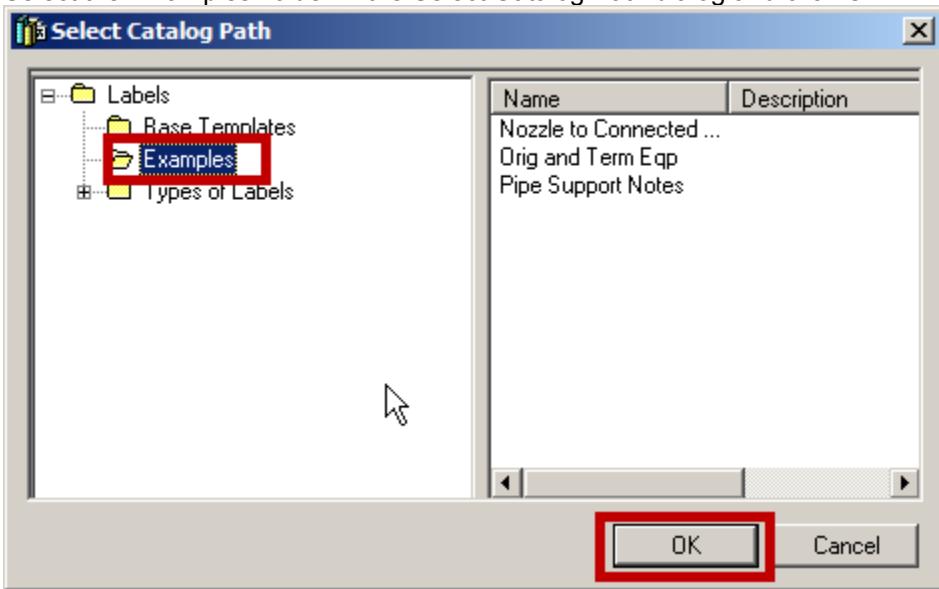
- Select 'Nozzle to Run' as the relationship and 'Piping Runs' as Related object type.



- Select the 'Name' property from the Standard category
- Change its Alias Name to 'Run Name'
- Click OK to save the label.
- Since we edited an existing label and changed its name, we are prompted to create a new label or rename existing. Select Yes to create a new label.



14. Select the 'Examples' folder in the Select Catalog Path dialog and click OK.



Using Filter Condition in Query to Return Multiple Objects

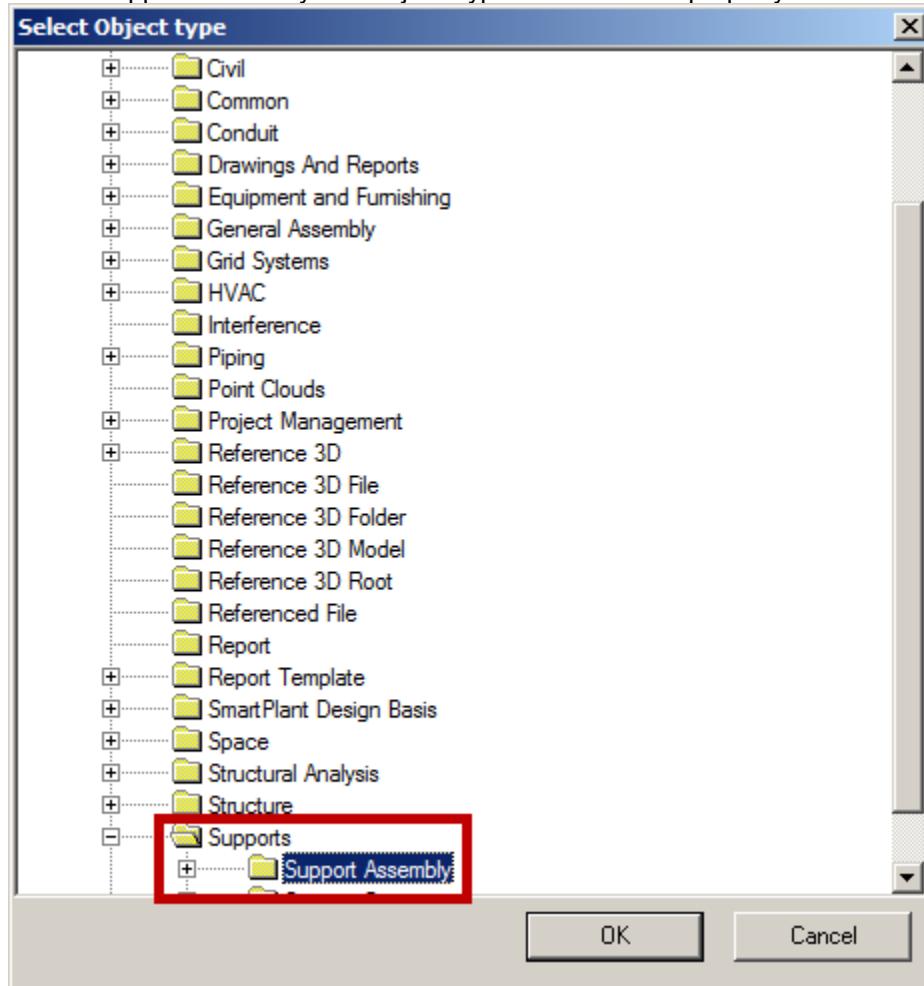
Objective: After this exercise, you will be able to create a label that returns multiple objects related to your target object

Using Filter="All"

This method is to be used when there are more than two related objects expected and all of them are to be returned by the query.

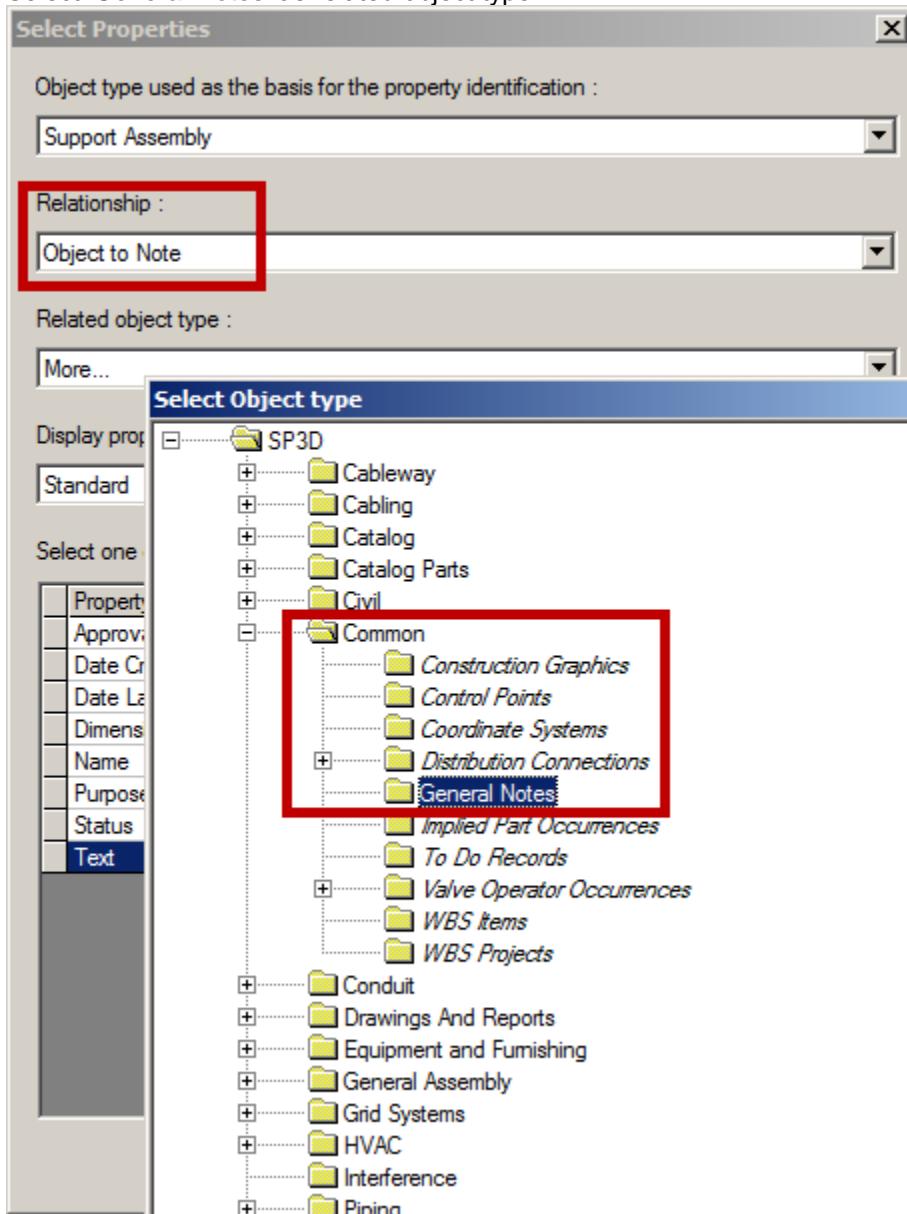
1. Switch to Catalog task
2. Tools → Define Label
3. Select the Examples folder and click 'New COM Label'
4. Name the label 'Pipe Support Notes'
5. Click the Add... button

6. Select 'Support Assembly' as 'Object Type to be used for property identification'

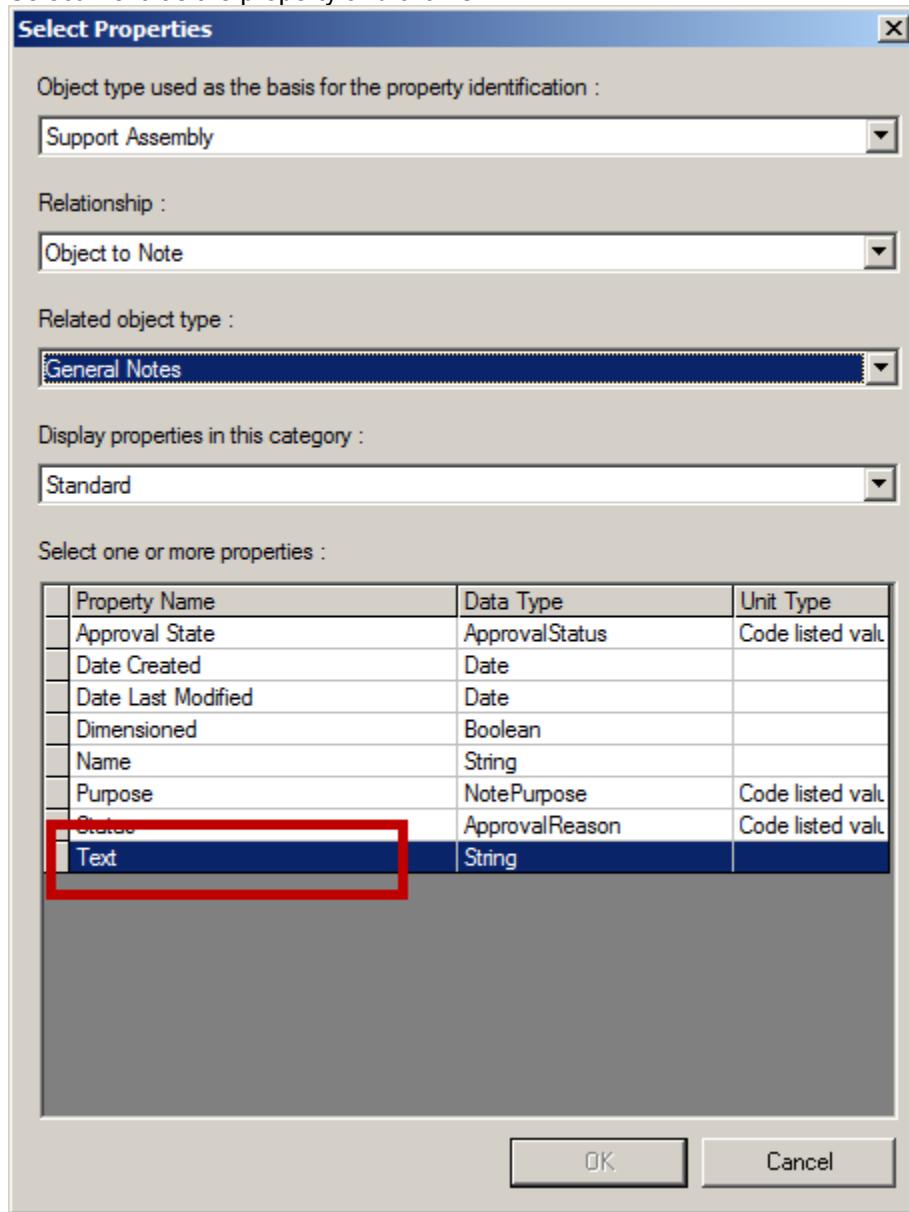


7. Select 'Object to Note' as the relationship

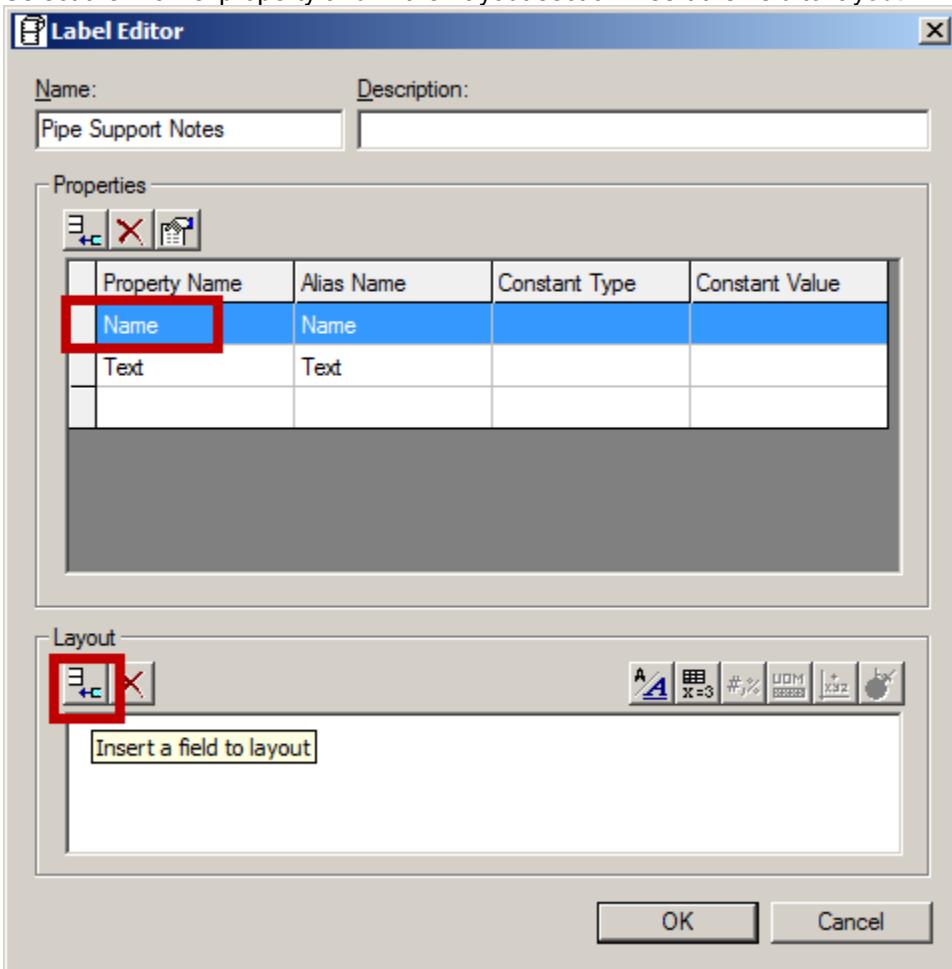
8. Select 'General Notes' as related object type



9. Select 'Text' as the property and click OK.

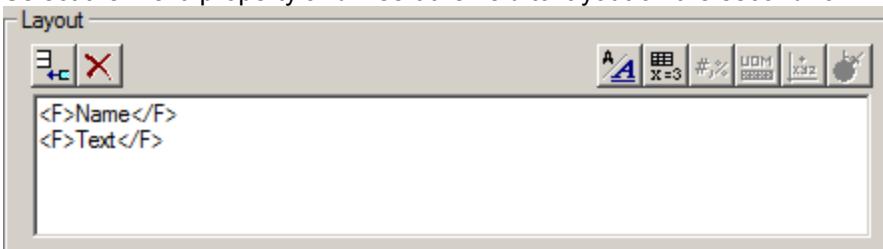


10. Select the 'Name' property and in the Layout section insert the field to layout



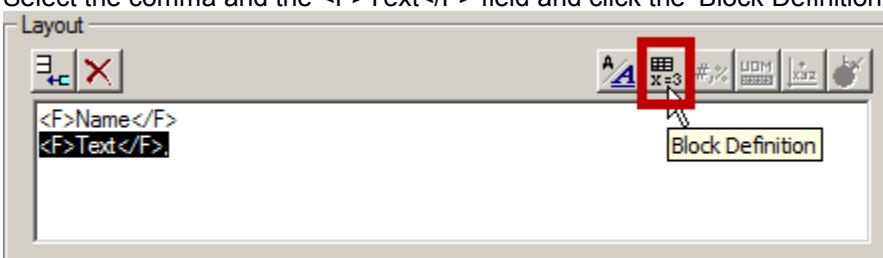
11. Press 'Enter' to go to a new line

12. Select the 'Text' property and insert the field to layout on the second row

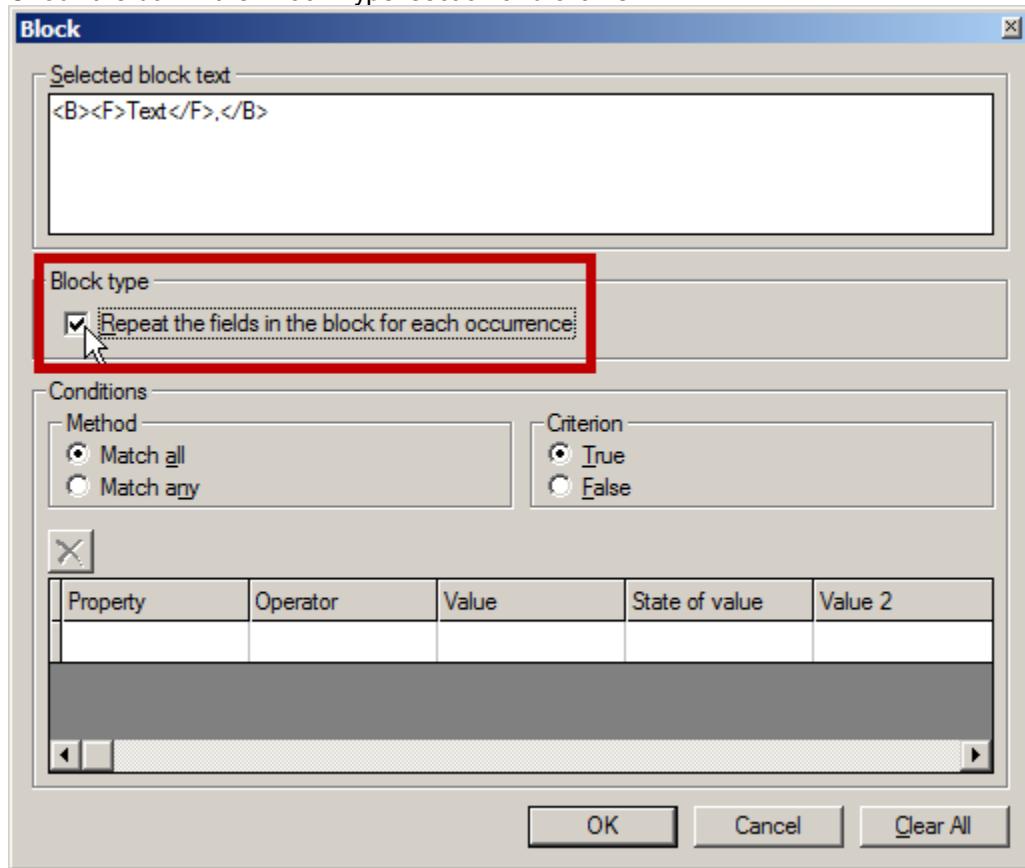


13. Enter a comma (,) after the <F>Text</F> field

14. Select the comma and the <F>Text</F> field and click the 'Block Definition' button



15. Check the box in the 'Block Type' section and click OK.



16. Click OK to save the label.

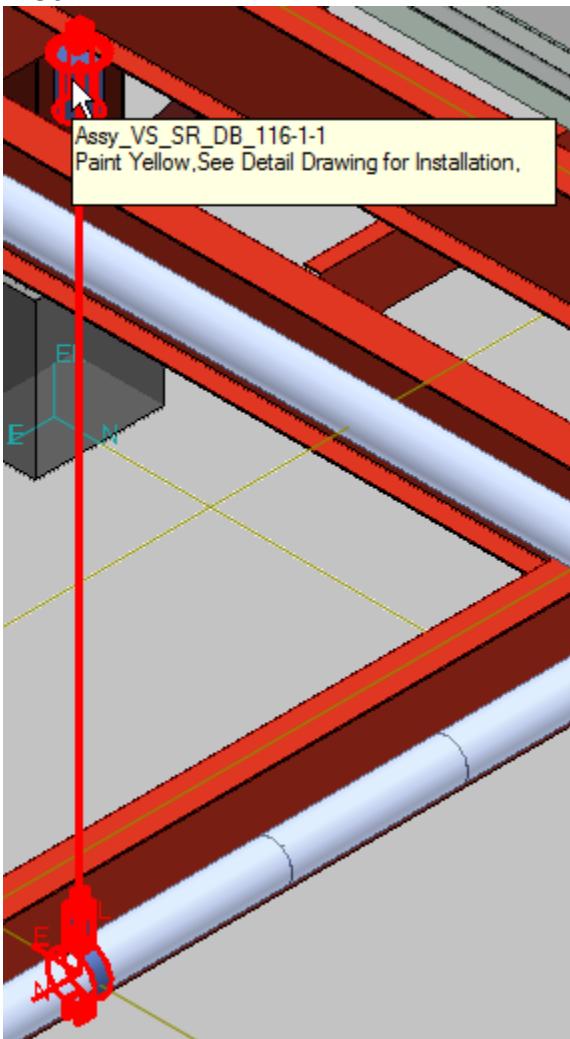
17. Edit the rqe file for the label and change the Filter="All"

```

<RETURNED_PROPERTY
  Name="Text"
  SQLType="Bstr">
  <PATHS>
    <PATH
      SourceType="IJDOobject"
      SourceBOC="SupportAssembly"
      DestinationInterface="IJGeneralNote"
      DestinationProperty="Text"
      DestinationBOC="GeneralNotes"
      Concatenate="No"
      PathSeparator="">
      <STROKES>
        <STROKE
          Interface="IJDOobject"
          RelationCollection="GeneralNote"
          Recursive="No"
          Filter="All"
          IsVirtualRelationship="No" />
      </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>

```

18. To test the label, use it as a tooltip for 'Support Assembly' and move your mouse over the support in U01.

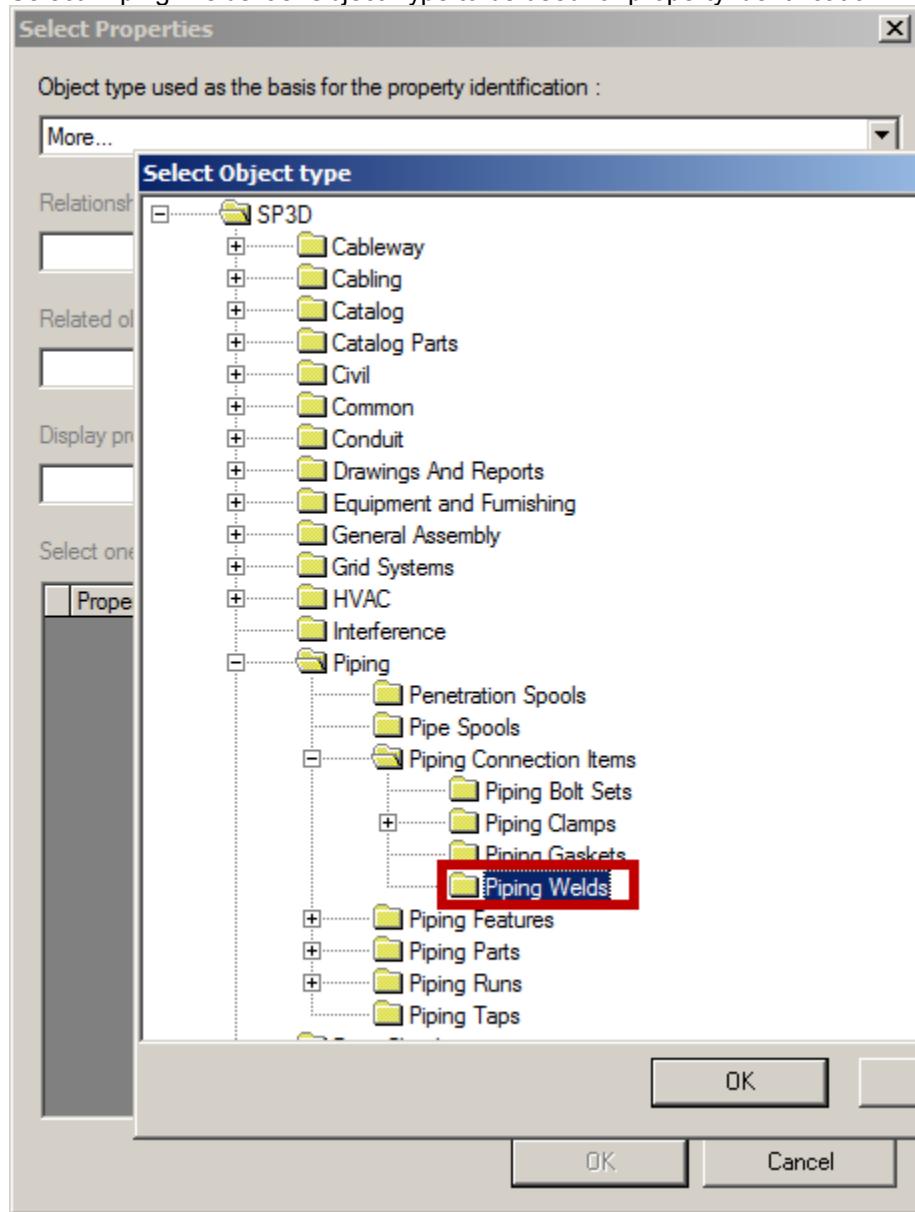


Using Filter="First" and "Last"

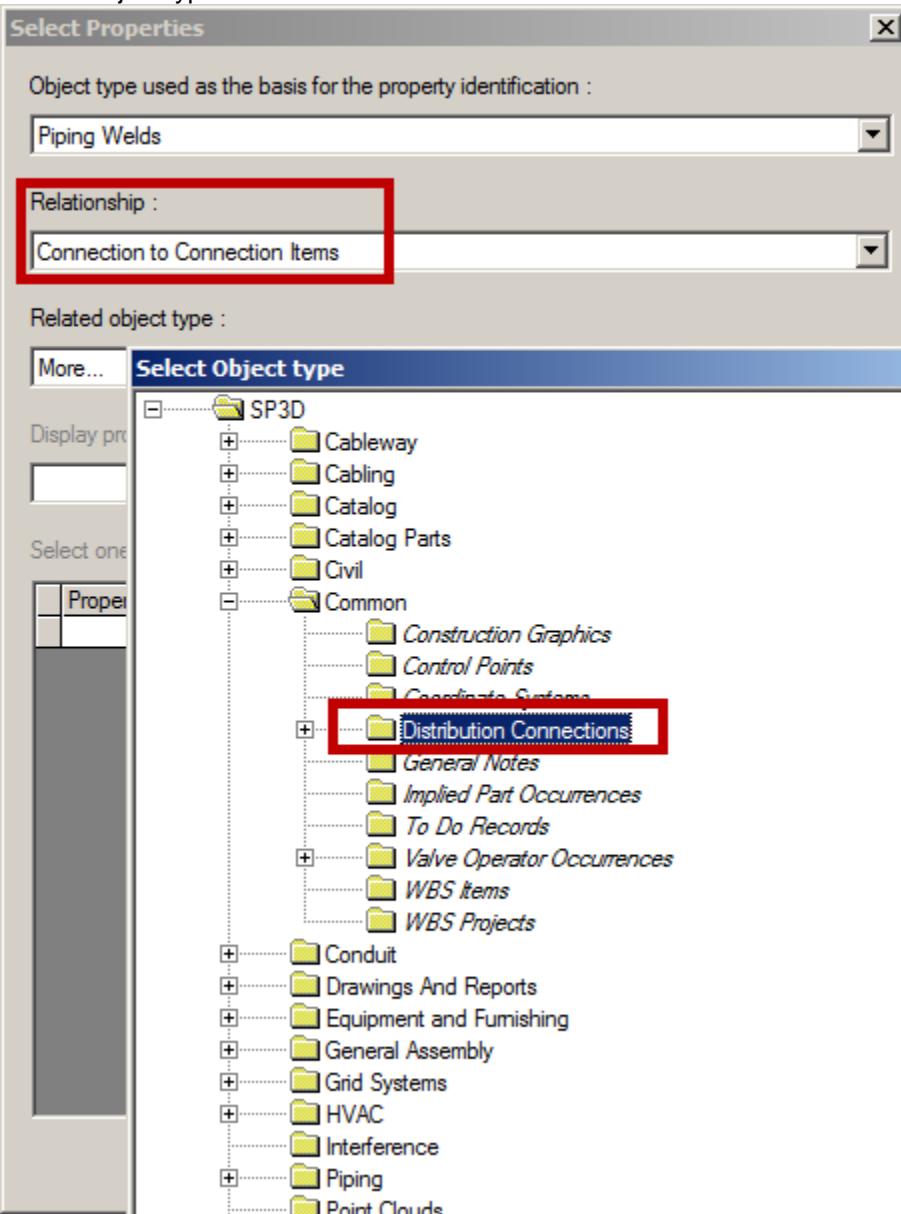
This method may be used when there are two objects and it is desired to return both objects independently

1. Switch to Catalog task
2. Tools → Define Label
3. Select the Examples folder and click 'New COM Label'
4. Name the label 'Weld Parents'
5. Delete the 'Name' property supplied by default.
6. Click the Add... button

7. Select 'Piping Welds' as 'Object Type to be used for property identification'

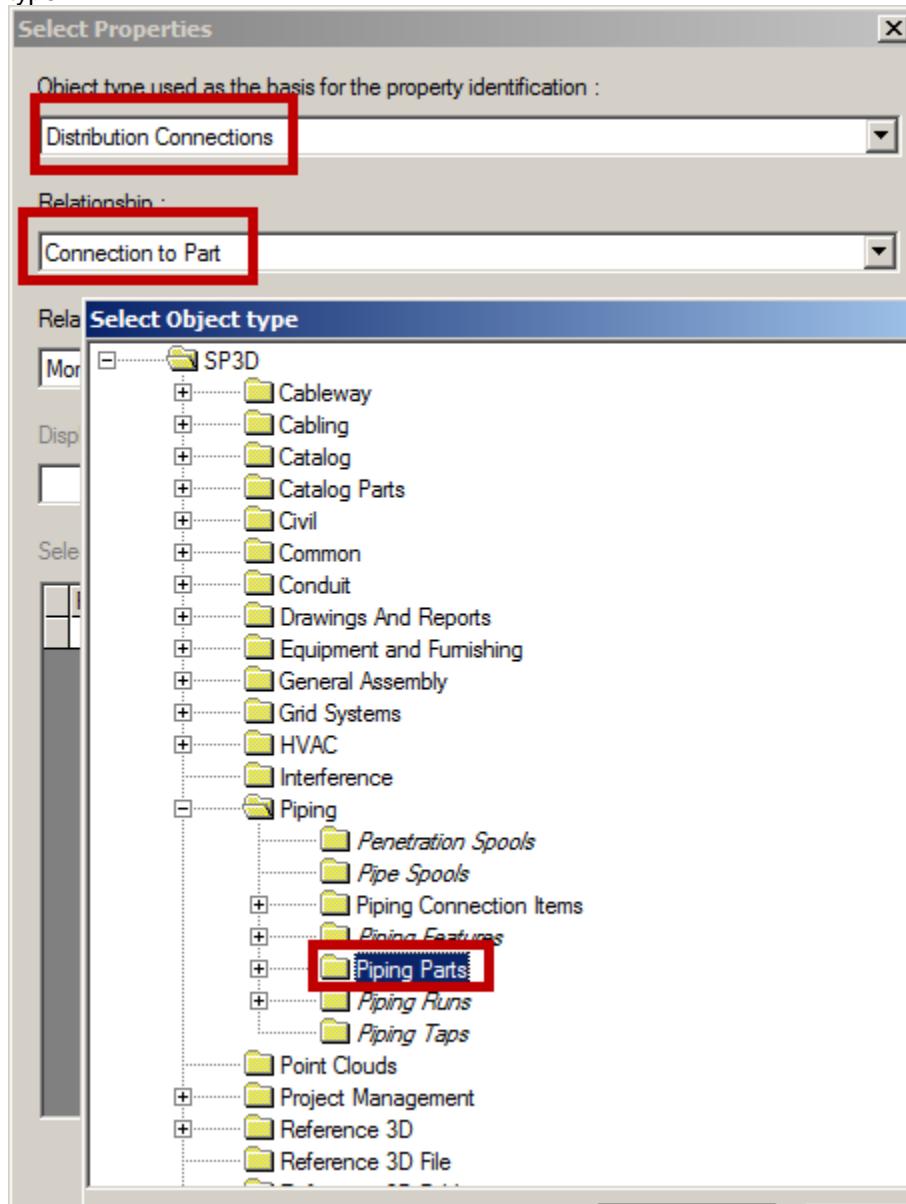


8. Select 'Connection to Connection Items' as the relationship and Distribution Connections as the related object type.



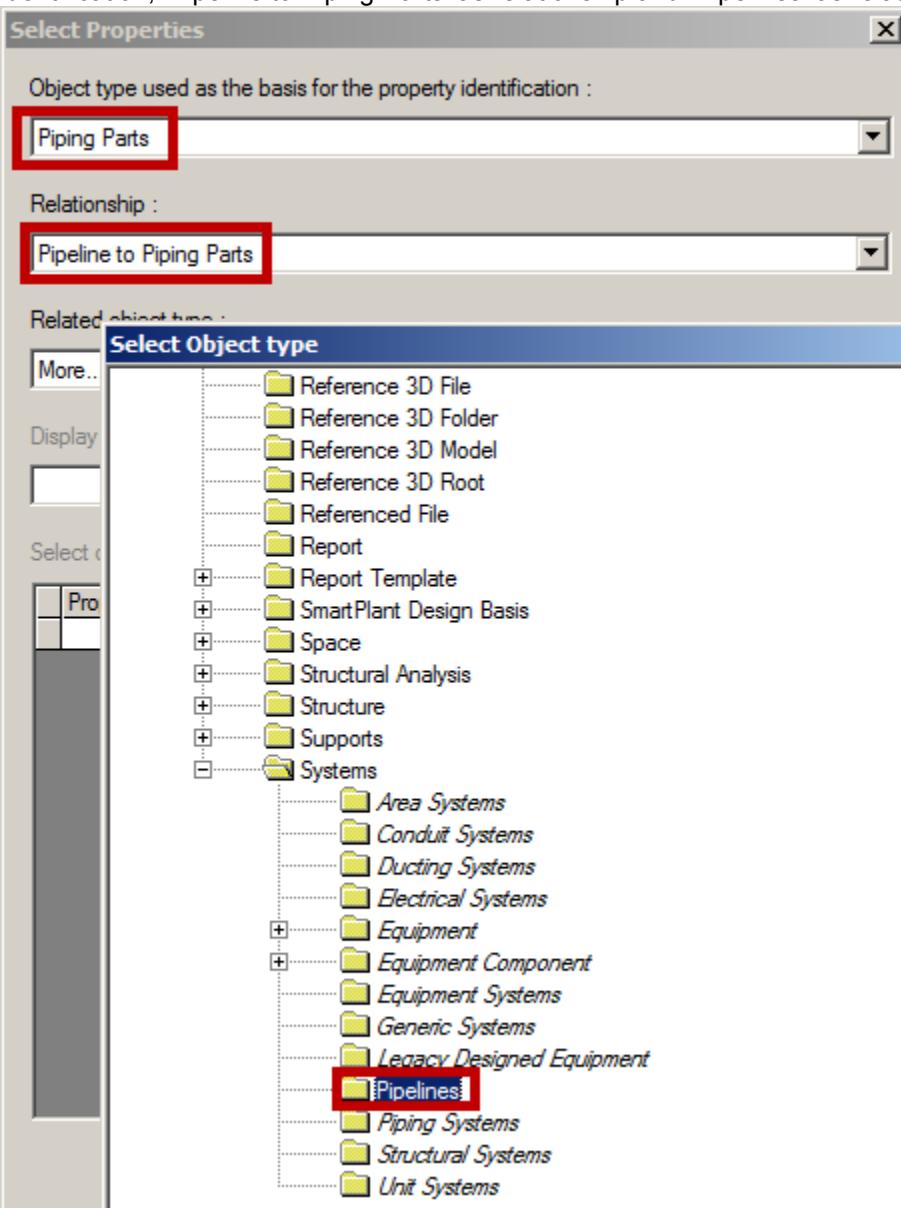
9. Pick 'Date Created' as the property. Note that we will only be using the STROKE from this relationship so it does not matter which property is selected.

10. Click Add... and add another property using 'Distribution Connection' as object type as basis for property identification, 'Connection to Part' as relationship and 'Piping Parts' as related object type.



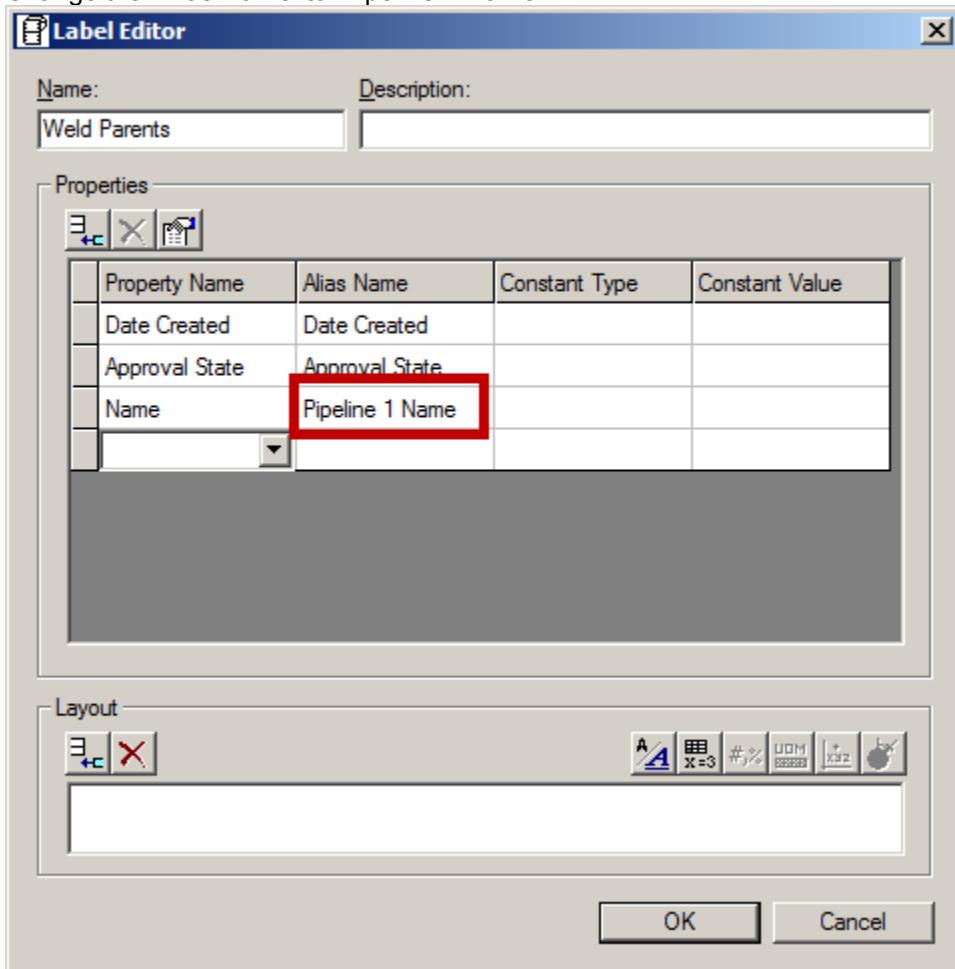
11. Pick the 'Approval State' property. As before, we will only be using the STROKE from this relationship so it does not matter which property is selected.

12. Click Add... and add another property using 'Piping Parts' as object type as basis for property identification, 'Pipeline to Piping Parts' as relationship and 'Pipelines' as related object type.



13. Pick the 'Name' property from standard category.

14. Change the 'Alias Name' to 'Pipeline 1 Name'



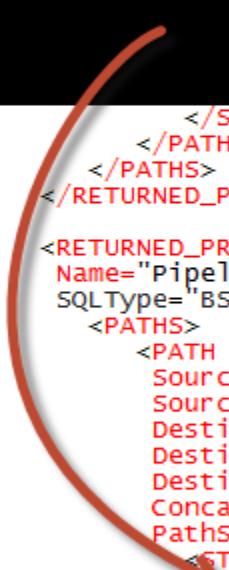
15. Click OK to create the label.

16. Edit the rqe file for the label.

17. Copy the STROKE from the second property 'Approval State' to the last property 'Pipeline 1 Name' (before the existing stroke) as shown below

```
<RETURNED_PROPERTY
  Name="Approval State"
  SQLType="Codelist">
  <PATHS>
    <PATH
      SourceType="IJDistribConnection"
      SourceBOC="DistribConnections"
      DestinationInterface="IJDOObject"
      DestinationProperty="Approvalstatus"
      DestinationBOC="PipingParts"
      Concatenate="No"
      PathSeparator="">
      <STROKES>
        <STROKE
          Interface="IJDistribConnection"
          RelationCollection="Parts"
          Recursive="No"
          Filter="First"
          IsVirtualRelationship="No" />
      </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>

<RETURNED_PROPERTY
  Name="Pipeline 1 Name"
  SQLType="Bstr">
  <PATHS>
    <PATH
      SourceType="IJRtePathGenPart"
      SourceBOC="PipingParts"
      DestinationInterface="IJNamedItem"
      DestinationProperty="Name"
      DestinationBOC="Pipelinesystems"
      Concatenate="No"
      PathSeparator="">
      <STROKES>
        <STROKE
          Interface="IJRtePathGenPart"
          RelationCollection="Pipeline"
          Recursive="No"
          Filter="First"
          IsVirtualRelationship="Yes" />
      </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>
</RETURNED_PROPERTIES>
```



18. Delete the Approval State returned_property from the rqe file

19. Copy the STROKE from the first property 'Date Created' to the last property 'Pipeline 1 Name' (before the existing stroke) as shown below

```
<RETURNED_PROPERTY
  Name="Date Created"
  SQLType="DBTimeStamp">
  <PATHS>
    <PATH
      SourceType="IJRteConnectionItem"
      SourceBOC="Pipewelds"
      DestinationInterface="IJDOObject"
      DestinationProperty="DateCreated"
      DestinationBOC="DistribConnections"
      Concatenate="No"
      PathSeparator="\\">
      <STROKES>
        <STROKE
          Interface="IJRteConnectionItem"
          RelationCollection="Connection"
          Recursive="No"
          Filter="First"
          IsVirtualRelationship="No"
          ExitValue="1" />
      </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>

<RETURNED_PROPERTY
  Name="Pipeline 1 Name"
  SQLType="Bstr">
  <PATHS>
    <PATH
      SourceType="IJRtePathGenPart"
      SourceBOC="PipingParts"
      DestinationInterface="IJNamedItem"
      DestinationProperty="Name"
      DestinationBOC="Pipelinesystems"
      Concatenate="No"
      PathSeparator="">
      <STROKES>
        <STROKE
          Interface="IJDistribconnection"
          RelationCollection="Parts"
          Recursive="No"
          Filter="First"
          IsVirtualRelationship="No" />
        <STROKE
          Interface="IJRtePathGenPart"
          RelationCollection="Pipeline"
          Recursive="No"
          Filter="First"
          IsVirtualRelationship="Yes" />
      </STROKES>
    </PATH>
  </PATHS>
</RETURNED_PROPERTY>
```

20. Delete the SourceType and SourceBOC from the 'Pipeline 1 Name' returned_property

```
<RETURNED_PROPERTY  
  Name="Pipeline 1 Name"  
  SQLType="BStr">  
  <PATHS>  
    <PATH  
      SourceType="IJRtePathGenPart"  
      SourceBOC="PipingParts"  
      DestinationInterface="IJNamedItem"  
      DestinationProperty="Name"  
      DestinationBOC="PipelineSystems"  
      Concatenate="No"  
      PathSeparator="">  
      <STROKES>  
        <STROKE  
          Interface="IJRteConnectionItem"  
          RelationCollection="Connection"  
          Recursive="No"  
          Filter="First"  
          IsVirtualRelationship="No"  
          ExitValue="1" />  
        <STROKE  
          Interface="IJDistribConnection"  
          RelationCollection="Parts"  
          Recursive="No"  
          Filter="First"  
          IsVirtualRelationship="No" />  
        <STROKE  
          Interface="IJRtePathGenPart"  
          RelationCollection="Pipeline"  
          Recursive="No"  
          Filter="First"  
          IsVirtualRelationship="Yes" />  
      </STROKES>  
    </PATH>  
  </PATHS>
```

21. Copy the SourceType and SourceBOC from the 'Date Created' returned_property to the 'Pipeline 1 Name' returned_property

```
<RETURNED_PROPERTY  
    Name="Date Created"  
    SQLType="DBTimeStamp">  
    <PATHS>  
        <PATH  
            SourceType="IJRteConnectionItem"  
            SourceBOC="Pipewelds"  
            DestinationInterface="IJDOObject"  
            DestinationProperty="DateCreated"  
            DestinationBOC="DistribConnections"  
            Concatenate="No"  
            PathSeparator="\\">  
            <STROKES>  
                <STROKE  
                    Interface="IJRteConnectionItem"  
                    RelationCollection="Connection"  
                    Recursive="No"  
                    Filter="First"  
                    IsVirtualRelationship="No"  
                    ExitValue="1" />  
            </STROKES>  
        </PATH>  
    </PATHS>  
</RETURNED_PROPERTY>  
  
<RETURNED_PROPERTY  
    Name="Pipeline 1 Name"  
    SQLType="Bstr">  
    <PATHS>  
        <PATH  
            DestinationInterface="IJNamedItem"  
            DestinationProperty="Name"  
            DestinationBOC="Pipelinesystems"  
            Concatenate="No"  
            PathSeparator="">  
            <STROKES>  
                <STROKE  
                    Interface="IJRteConnectionItem"  
                    RelationCollection="Connection"  
                    Recursive="No"  
                    Filter="First"  
                    IsVirtualRelationship="No"  
                    ExitValue="1" />  
                <STROKE  
                    Interface="IJDistribConnection"  
                    RelationCollection="Parts"  
                    Recursive="No"  
                    Filter="First"  
                    IsVirtualRelationship="No" />  
                <STROKE  
                    Interface="IJRtePathGenPart"  
                    RelationCollection="Pipeline"  
                    Recursive="No"  
                    Filter="First"  
                    IsVirtualRelationship="Yes" />  
            </STROKES>
```

22. Delete the Date Created returned_property from the rqe file

23. Copy the entire 'Pipeline 1 Name' RETURNED_PROPERTY and paste it right below itself, then change its Name to 'Pipeline 2 Name'

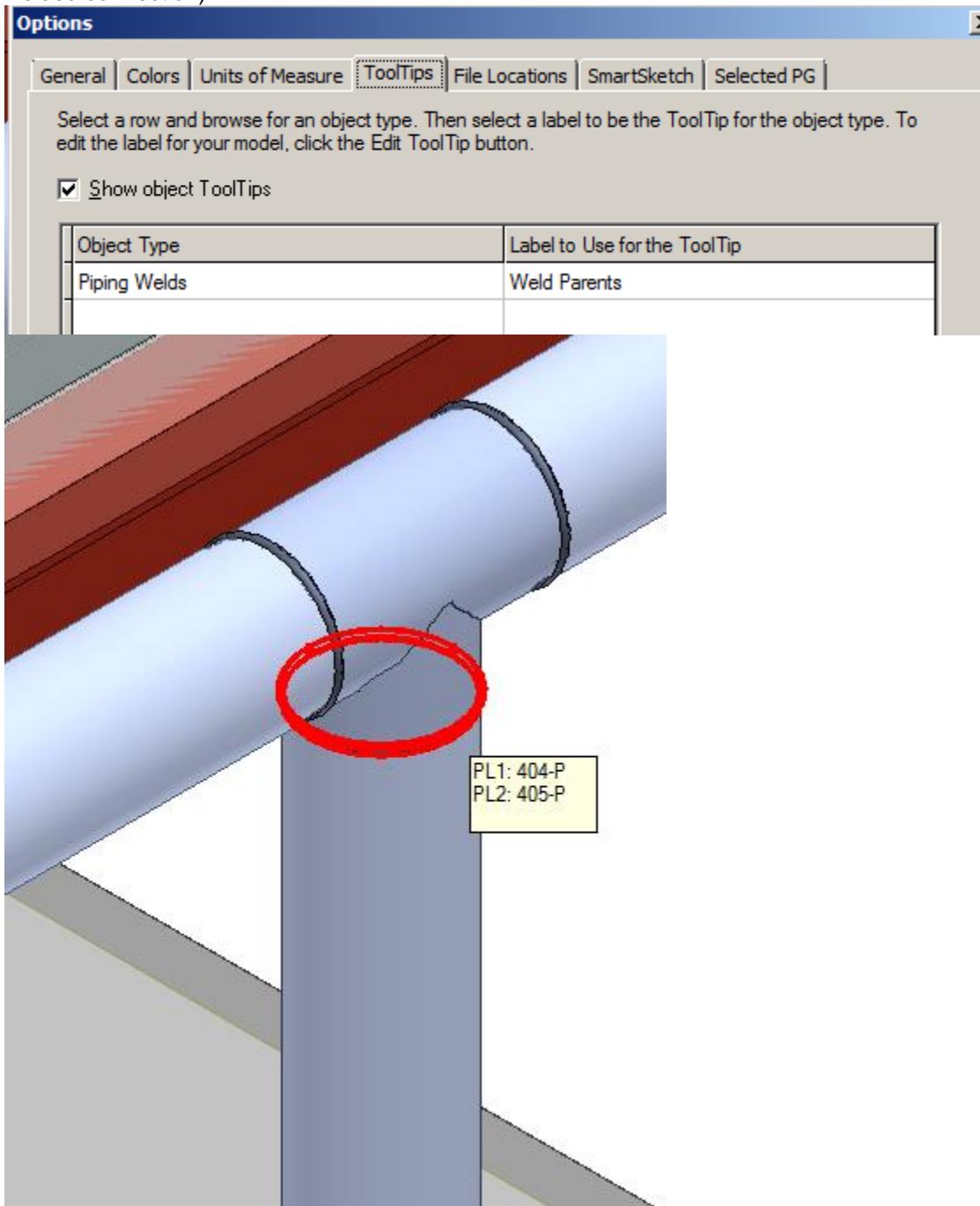
```
<RETURNED_PROPERTIES>
  <RETURNED_PROPERTY Name="Pipeline 1 Name" SQLType="BStr">
    <PATHS>
      <PATH SourceType="IJRteConnectionItem" SourceBOC="Pipewell">
        <STROKES>
          <STROKE Interface="IJRteConnectionItem" RelationCol1="Pipewell">
            <STROKE Interface="IJDistribConnection" RelationCol1="Distrib">
              <STROKE Interface="IJRtePathGenPart" RelationCollection="Path">
                </STROKES>
              </PATH>
            </PATHS>
          </RETURNED_PROPERTY>
        <RETURNED_PROPERTY Name="Pipeline 2 Name" SQLType="BStr">
          <PATHS>
            <PATH SourceType="IJRteConnectionItem" SourceBOC="Pipewell">
              <STROKES>
                <STROKE Interface="IJRteConnectionItem" RelationCol1="Pipewell">
                  <STROKE Interface="IJDistribConnection" RelationCol1="Distrib">
                    <STROKE Interface="IJRtePathGenPart" RelationCollection="Path">
                      </STROKES>
                    </PATH>
                  </PATHS>
                </RETURNED_PROPERTY>
              </RETURNED_PROPERTIES>
            ,
```

24. In the 'Pipeline 2 Name' property, change the Filter="Last" for the second STROKE

```
<RETURNED_PROPERTY  
  Name="Pipeline 2 Name"  
  SQLType="Bstr">  
  <PATHS>  
    <PATH  
      SourceType="IJRteConnectionItem"  
      SourceBOC="Pipewelds"  
      DestinationInterface="IJNamedItem"  
      DestinationProperty="Name"  
      DestinationBOC="Pipelinesystems"  
      Concatenate="No"  
      PathSeparator="">  
      <STROKES>  
        <STROKE  
          Interface="IJRteConnectionItem"  
          RelationCollection="Connection"  
          Recursive="No"  
          Filter="First"  
          IsVirtualRelationship="No"  
          ExitValue="1" />  
        <STROKE  
          Interface="IJDistribConnection"  
          RelationCollection="Parts"  
          Recursive="No"  
          Filter="Last"  
          IsVirtualRelationship="No" />  
        <STROKE  
          Interface="IJRtePathGenPart"  
          RelationCollection="Pipeline"  
          Recursive="No"  
          Filter="First"  
          IsVirtualRelationship="Yes" />  
      </STROKES>  
    </PATH>  
  </PATHS>  
</RETURNED_PROPERTY>
```

25. Edit the 'Weld Parents' label in the catalog task. Two properties are shown
26. In the 'Layout' section, key in PL1: and then insert 'Pipeline 1 Name' to layout
27. Press Enter and key in PL2: and then insert 'Pipeline 2 Name' to layout
28. Click OK to save the label

29. Test the label as a tooltip in U04 pipelines 404-P and 405-P (these are two pipelines that join at a welded connection)



30. For extra credit, use conditional formatting to only output PL2 if its value is different from PL1. You can do this by putting a conditional block around PL2.

Creating a label that returns unit name for piping parts and piping welds

Objective: This exercise is intended to consolidate the learning in the previous exercises to combine into a single label the concepts of multiple strokes in a path, multiple paths with different source types, combining multiple queries into a single query, recursive navigation using the Implements condition, using edges etc. Step by step instructions will not be provided. The label created should be named 'Examples\Unit Name' and the alias name for the property should be 'Unit Name'

SQL Queries

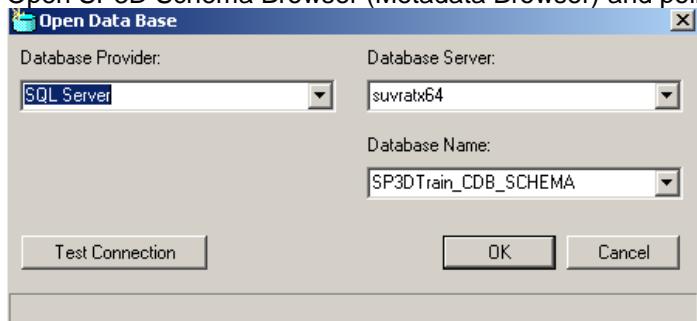
Writing SQL Queries

Objective: After this exercise, you will be able to

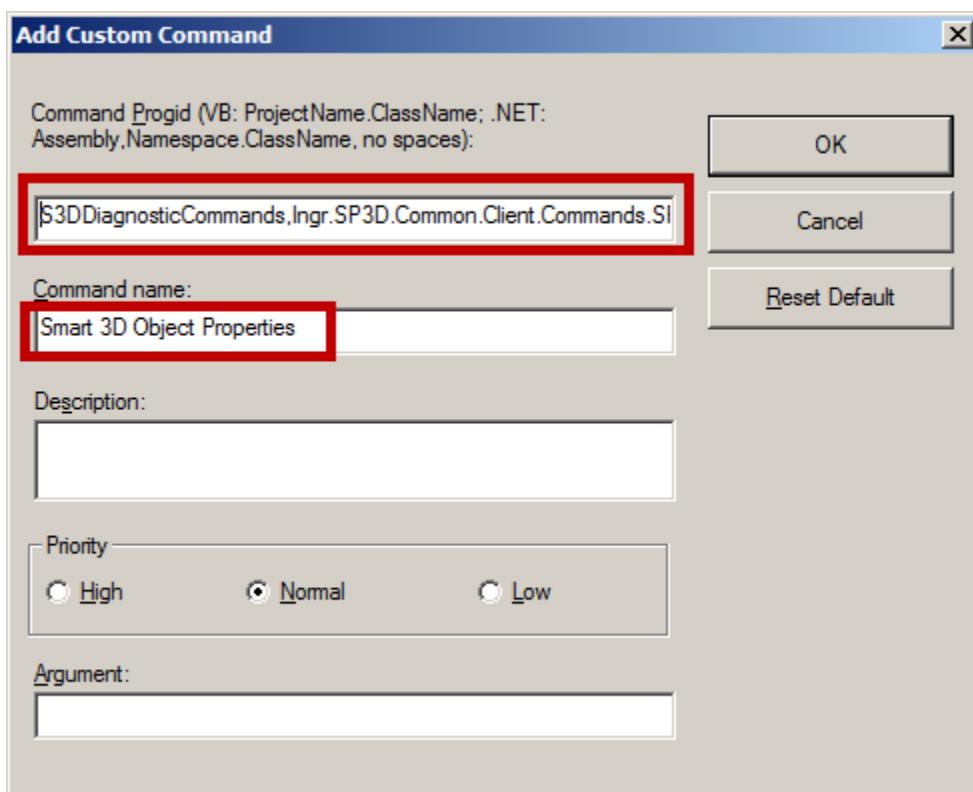
- Write simple SQL queries for direct properties
- Output values and query on codelisted properties

Direct Property Queries

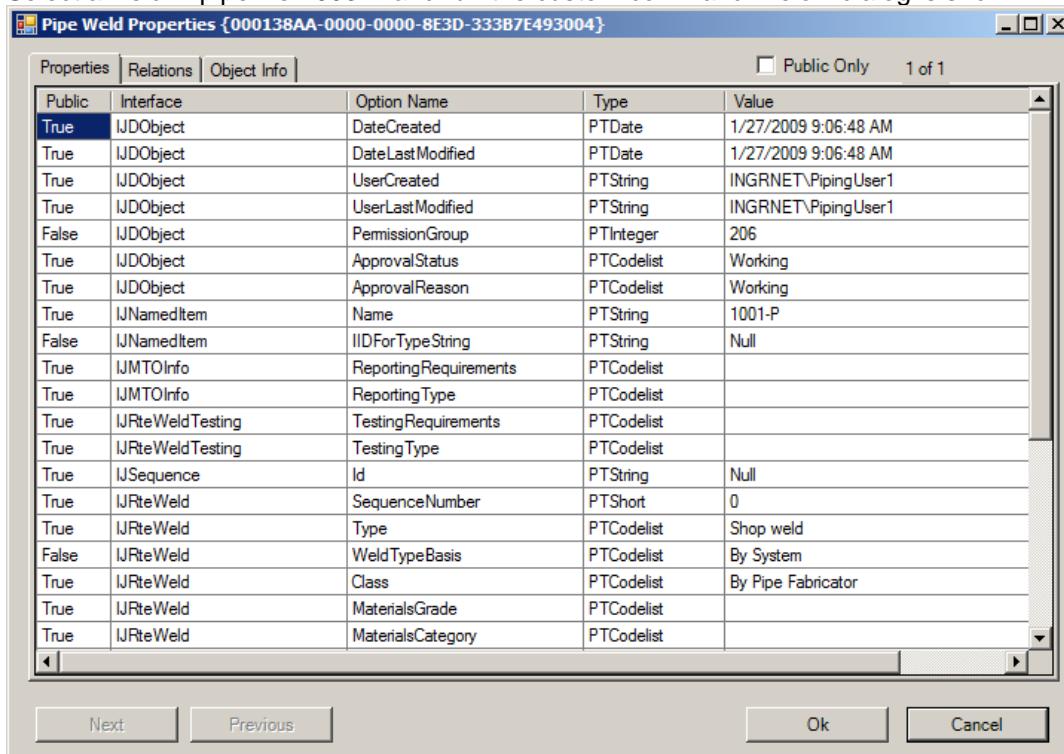
1. Open SP3D Schema Browser (Metadata Browser) and point to your catalog schema



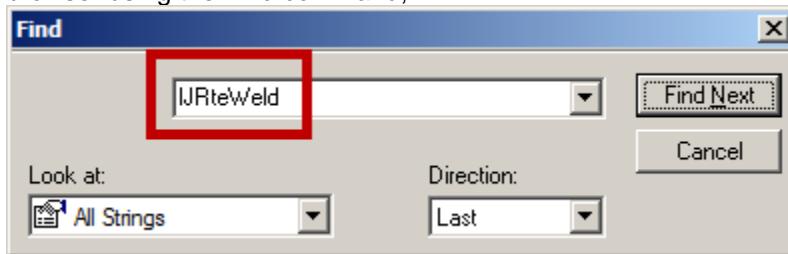
2. In Smart 3D, define workspace to U02
3. Using Tools → Custom Commands – Add... add a custom command with ProgID S3DDiagnosticCommands,Ingr.SP3D.Common.Client.Commands.SP3DObjectProperties



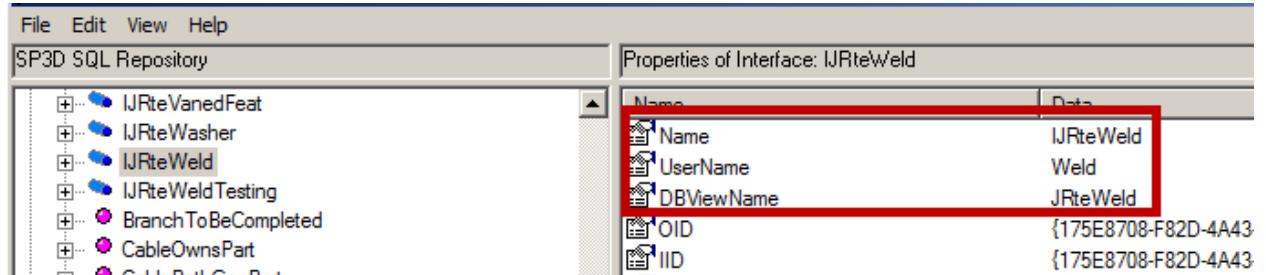
4. This command is an alternative to using the Repository Browser and is available in Smart 3D 2014 and later versions.
5. Select a weld in pipeline 2003-P and run the custom command. Below dialog is shown



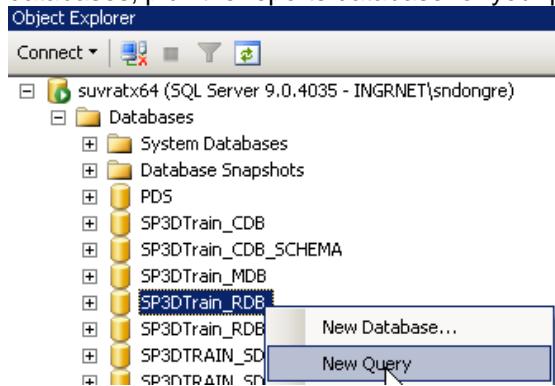
6. This shows us that IJRteWeld interface is implemented by this object and there will be an entry for this object in the database view corresponding to this interface
7. To find the view name related to this interface, locate the IJRteWeld interface in the schema browser using the Find command,



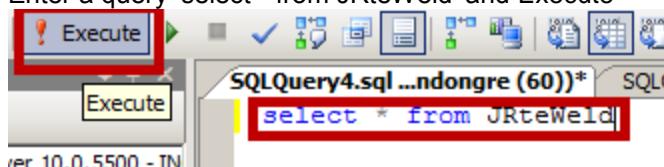
8. This shows us that the DBViewName is JRteWeld



9. Open Microsoft SQL Server Management Studio, make a connection to your server, expand databases, pick the reports database for your plant and select New Query.



10. Enter a query 'select * from JRteWeld' and Execute



Codelisted attribute queries

11. Notice that the value for 'Type' is a number. This is because it is a codelisted attribute.

SQLQuery4.sql ...ndongre (60)* SQLQuery2.sql ...ndongre (56)*				
Results Messages				
Oid	Type	WeldTypeBasis	Value	
1 000138AA-0000-0000-1A02-06D7BF4F1F04	5	5		
2 000138AA-0000-0000-2A03-06D7BF4F1F04	5	5		

12. Notice that the codelist is resolved in the custom command dialog.

Properties Relations Object Info Public Only				
Public	Interface	Option Name	Type	Value
True	IJRteWeldTesting	TestingType	PTCodelist	
True	IJSequence	Id	PTString	Null
True	IJRteWeld	SequenceNumber	PTShort	0
True	IJRteWeld	Type	PTCodelist	Shop weld

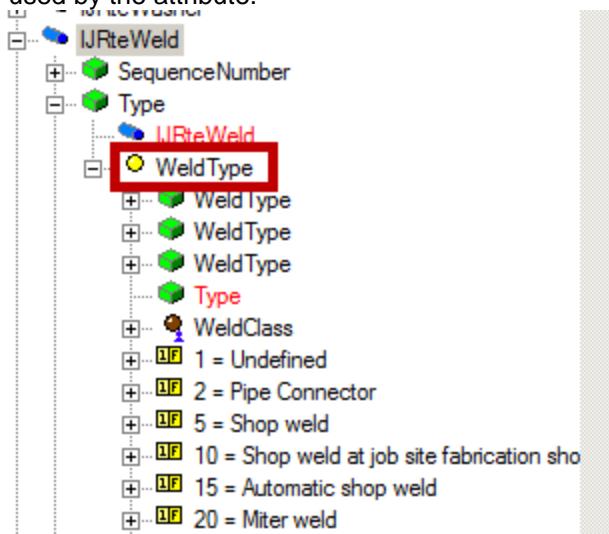
13. There are two ways to resolve codelists via SQL query. Some views have an additional view that resolves codelists, typically named *ViewName_CL*. This view is one of those kinds

14. Enter a query 'select * from JRteWeld_CL' and Execute.

15. Notice that now, both the number and the short and long value are returned by the query

SQLQuery4.sql ...ndongre (60)* SQLQuery2.sql ...ndongre (56)* Object Explorer Details				
Results Messages				
Oid	Type	Type_ShortValue	Type_LongValue	
1 000138AA-0000-0000-1A02-06D7BF4F1F04	5	Shop weld	Shop weld	
2 000138AA-0000-0000-2A03-06D7BF4F1F04	5	Shop weld	Shop weld	
3 000138AA-0000-0000-2C02-06D7BF4F1F04	5	Shop weld	Shop weld	

16. In case a special view does not exist, codelists can be resolved using codelist specific views. To find the codelist specific view, expand the interface in the schema browser to see which codelist is used by the attribute.



17. By convention, views for codelists are named CL_CodelistName

18. Enter a query 'select * from CL_WeldType' and Execute

	ValueID	ShortStringValue	LongStringValue
1	1	Undefined	Undefined
2	2	Pipe Connector	Pipe Connector
3	5	Shop weld	Shop weld
4	10	Shop weld at job site fabrication shop	Shop weld at job site fabrication shop

19. To define the query using the original and the codelist views, we must join the two together. The value for the Type attribute in JRteWeld view corresponds to the ValueID in the CL_WeldType view. So enter a query as below and Execute.

```

SQLQuery4.sql ...ndongre (60)* SQLQuery2.sql ...ndongre (56)* Object Explorer Details
select * from JRteWeld
  join CL_WeldType on JRteWeld.Type = CL_WeldType.ValueID
  
```

20. By scrolling to the right side in the results pane, you can see that the resolved values are available to you.

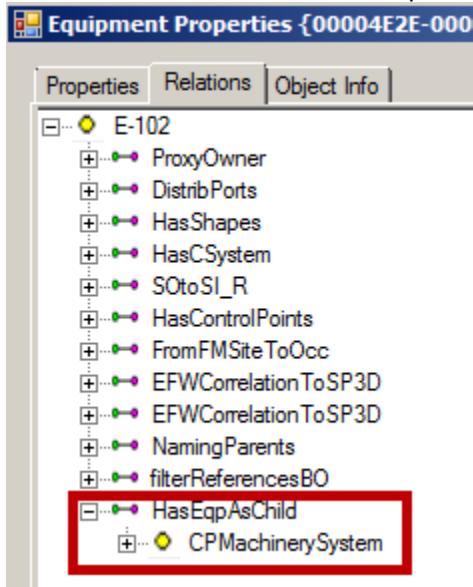
Create SQL Label to return name of Equipment Parent

Objective: After this lab, you will be able to

- Trace relationships using schema browser
- Define a SQL label query for equipment to system relationship
- Optimize the query to remove unnecessary joins
- Use substring functions to further refine the returned values

Finding relationship between equipment and parent

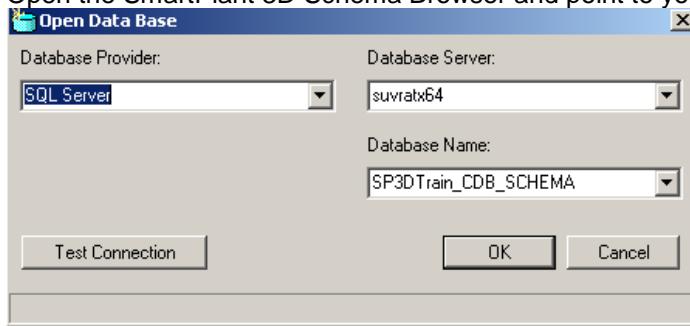
1. Define workspace to be U01
2. Select the equipment E-102 and start the 'Smart 3D Object Properties' command
3. Switch to the Relations tab and expand the HasEqpAsChild relationship.



4. Select CPMachinerySystem and click the GoTo button. Now the information shown is for the parent system and the Previous button is active. You can go between the equipment and the system using these buttons.

Tracing relationship in schema browser

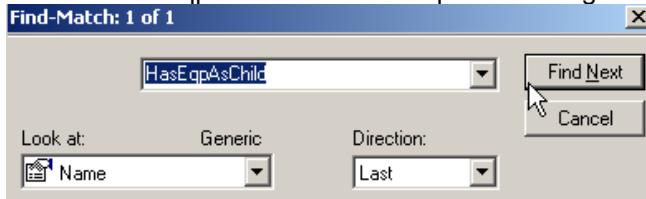
1. Open the SmartPlant 3D Schema Browser and point to your catalog schema



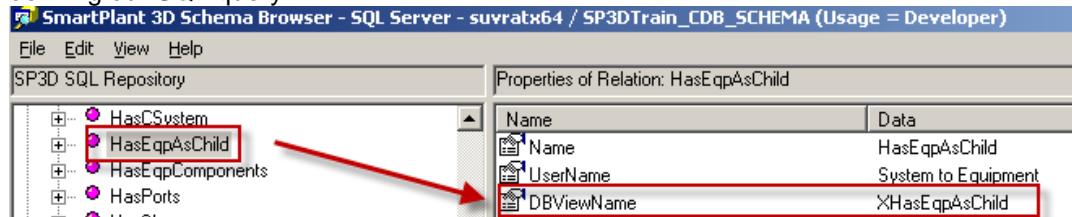
2. Use Find command



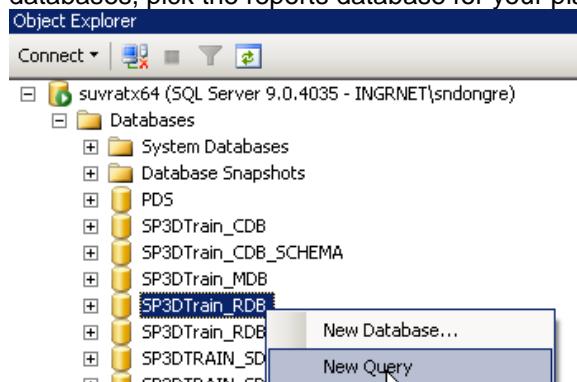
3. Look for 'HasEqpAsChild' relationship name using 'Name' as the 'Look at' criterion.



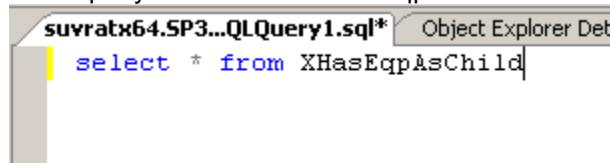
4. Software finds the named relationship. In the right pane, we can see name of database view for defining our SQL query.



5. Open Microsoft SQL Server Management Studio, make a connection to your server, expand databases, pick the reports database for your plant and select New Query.



6. Enter query 'select * from XHasEqpAsChild' and click 'Execute'



7. This will return the oid of the two related objects

8. Since we want names of related objects, we will make a join with the view for names (JNamedItem). Enter SQL query as below and execute

```
select * from XHasEqpAsChild x1
join JNamedItem j1 on x1.oidorigin = j1.oid
join JNamedItem j2 on x1.oiddestination = j2.oid
```

9. This gives us below result and tells us that for this relationship, system is on the origin side (j1) and equipment on destination side (j2).

	DidOrigin	DidDestination	RelationName	Id	ItemName	IIDForTypeString	Id	ItemName
1	0003454-0000-0000-1504-030071496404	0004E2E-0000-0000-3E00-31A878453804		00033454-0000-0000-1504-030071496404	Equipment	98EBE398-7F03-11D4-A205-001048364684	0004E2E-0000-0000-3E00-31A878453804	T-101
2	0003454-0000-0000-1504-030071496404	0004E2E-0000-0000-3C00-A4CE70493804		00033454-0000-0000-1504-030071496404	Equipment	98EBE398-7F03-11D4-A205-001048364684	0004E2E-0000-0000-3C00-A4CE70493804	P-101
3	0003454-0000-0000-1504-030071496404	0004E2E-0000-0000-E004-A4CE70493804		00033454-0000-0000-1504-030071496404	Equipment	98EBE398-7F03-11D4-A205-001048364684	0004E2E-0000-0000-E004-A4CE70493804	B-101

10. Since we are interested in the name of a parent system given an equipment, we will make the following change to query

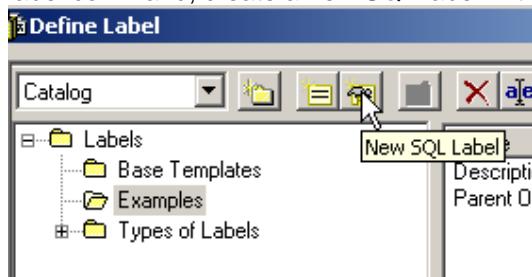
```
select j1.Itemname from XHasEqpAsChild x1
join JNamedItem j1 on x1.oidorigin = j1.oid
join JNamedItem j2 on x1.oiddestination = j2.oid
where j2.Itemname = 'E-102'
```

11. To make the query generic, we will change it to ask for OID of the object on source side

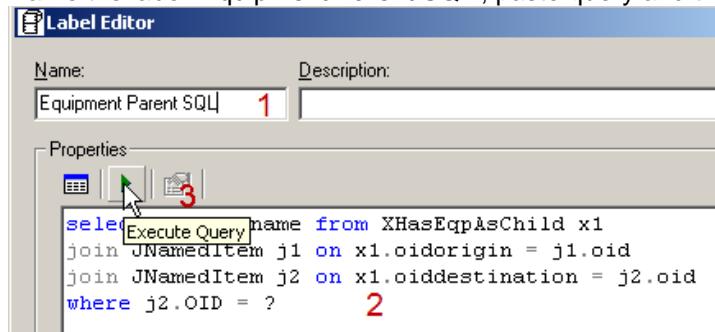
```
select j1.Itemname from XHasEqpAsChild x1
join JNamedItem j1 on x1.oidorigin = j1.oid
join JNamedItem j2 on x1.oiddestination = j2.oid
where j2.OID = ?
```

Defining a SQL Label

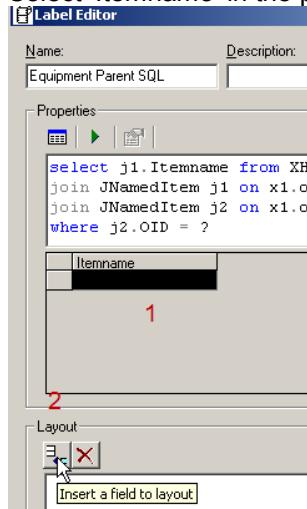
12. Now we will define an SQL label inside SP3D. Switch to Catalog task and using Tools – Define label command, create a new SQL label in the Examples folder.



13. Name the label 'Equipment Parent SQL', paste query and then click 'Execute Query'



14. Select 'Itemname' in the pane that appears and click 'Insert a field to layout'

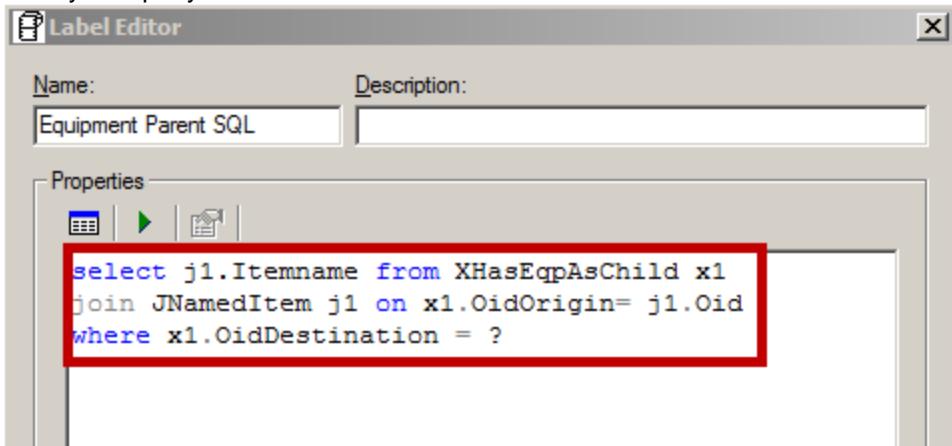


15. Click 'OK' to save the label. Dismiss a message if it appears.
16. Test the label as a tooltip in the Equipment task.

Optimizing the SQL Query and Label

In our query, we made two joins to the JNamedItem view to get the names of the equipment and its parent. However in our label, we only returned the name of the parent (from view with alias j1). While the join to j2 (equipment name) was useful during the creation phase, it is no longer necessary for execution of the label. Let us edit the query to remove the join.

1. Switch to the Catalog task and edit the 'Equipment Parent SQL' label
2. Modify the query to be as below

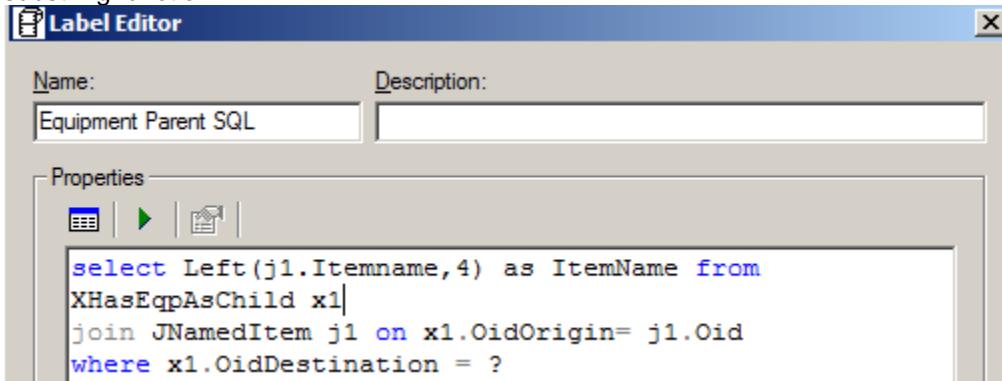


3. Click OK to save the label.
4. Exit and reenter the session, saving if prompted.
5. Test the label as tooltip in equipment task. Notice that it gives exactly the same result as before.

Returning Substring of a property

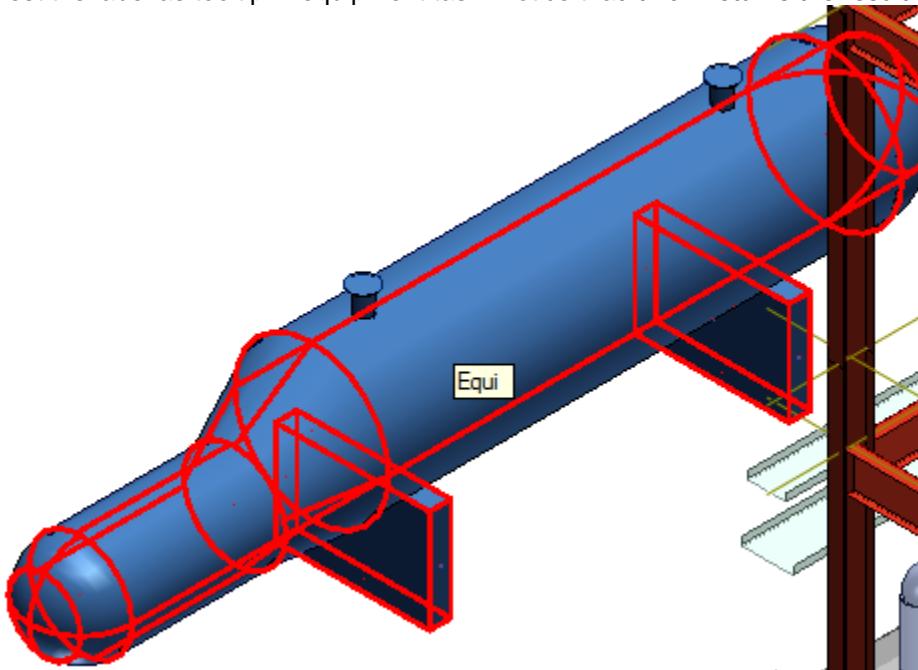
There are occasions when it is desired to query on the substring of a property or return the substring of a property. The only way to do this (without writing code) is using SQL queries. Let us edit our label to only return the first four characters of the name of the parent system.

1. Switch to the Catalog task and edit the 'Equipment Parent SQL' label
2. Modify the query to be as below – notice that we needed to provide an Alias when we used a substring function



3. Click OK to save the label.
4. Exit and reenter the session, saving if prompted.

5. Test the label as tooltip in equipment task. Notice that it now returns the result 'Equi'



Define SQL Filter

For particularly complex filters or for filters where an edge is not available, a SQL filter may be defined. The rules for a SQL filter are that it must return OIDs and that there must not be a ? in the query. Let us define a SQL filter that returns the equipment whose parent has a name whose name starts with E and contains the letters Eqp elsewhere.

1. In SQL Management Studio, enter query as below and Execute.

```
SQLQuery5.sql ...ndongre (57)* SQLQuery4.sql ...ndongre (60))*
select * from XHasEqpAsChild x1
| join JNamedItem j1 on j1.Oid = x1.OidOrigin
| join JNamedItem j2 on j2.Oid = x1.OidDestination
| where j1.ItemName like 'E%Eqp'
```

2. This query returns several rows. To select only the children, we can modify the query as below and execute it.

```
select j2.Oid from XHasEqpAsChild x1
| join JNamedItem j1 on j1.Oid = x1.OidOrigin
| join JNamedItem j2 on j2.Oid = x1.OidDestination
| where j1.ItemName like 'E%Eqp'
```

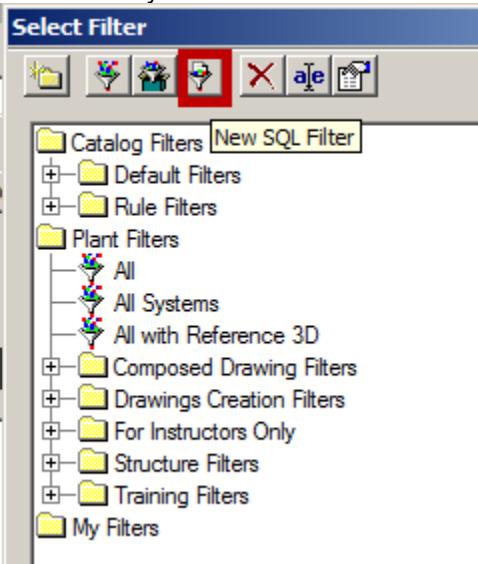
3. This query returns oids of the equipment items which satisfy our desired condition. We can further optimize this query by removing the join to JNamedItem for the equipment names since they are not returned in the query results or part of the where clause.

```
select x1.OidDestination from XHasEqpAsChild x1
| join JNamedItem j1 on j1.Oid = x1.OidOrigin
| where j1.ItemName like 'E%Eqp'
```

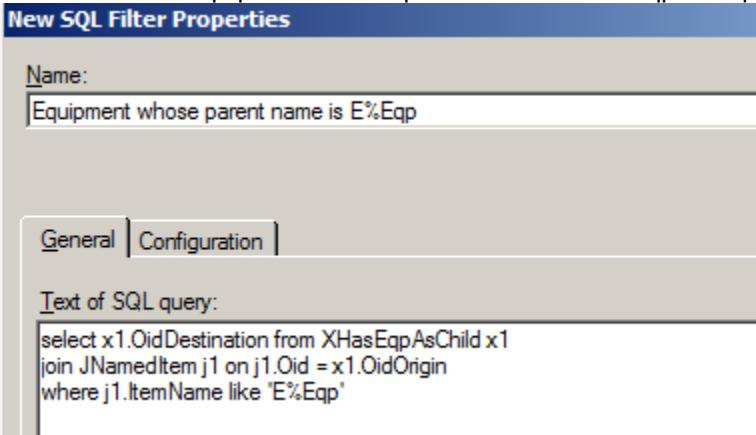
4. Select the query in SQL Management Studio and copy to clipboard

5. In Smart 3D, define workspace and select More... on the dialog

- Select the My Filters folder and click the 'New SQL Filter' command.



- Name the filter 'Equipment whose parent name is E%Eqp' and paste in the query from clipboard



- Click OK to define the filter.
- Select the filter just defined and click OK to define workspace.

Creating SQL Reports

Objective: After this exercise, you will be able to create a report that uses SQL queries and filter based SQL queries. You will also be able to convert the report into one suitable for embedding in a drawing.

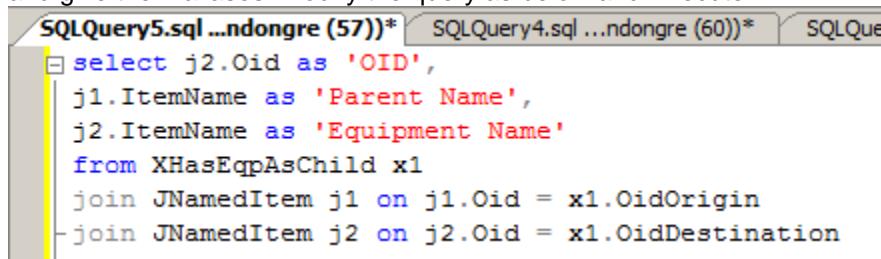
SQL Query Report

- In SQL Server Management Studio, enter the query as shown below and Execute

The screenshot shows an open query window in SQL Server Management Studio. The window title is 'SQLQuery5.sql ...ndongre (57)*'. The query itself is:

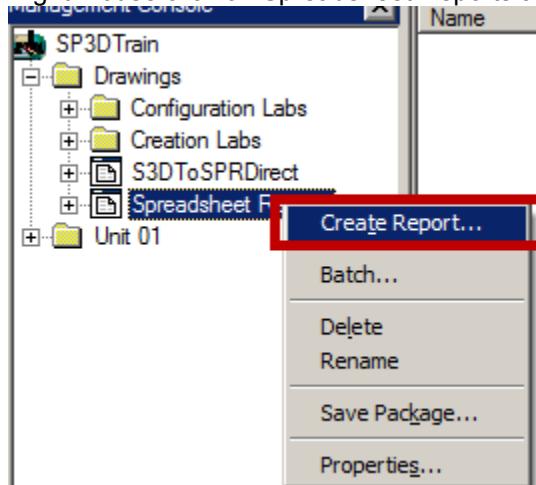
```
select * from XHasEqpAsChild x1
join JNamedItem j1 on j1.Oid = x1.OidOrigin
join JNamedItem j2 on j2.Oid = x1.OidDestination
```

2. Notice that the query returns several columns, we will identify the ones that we want in our report and give them aliases. Modify the query as below and Execute.

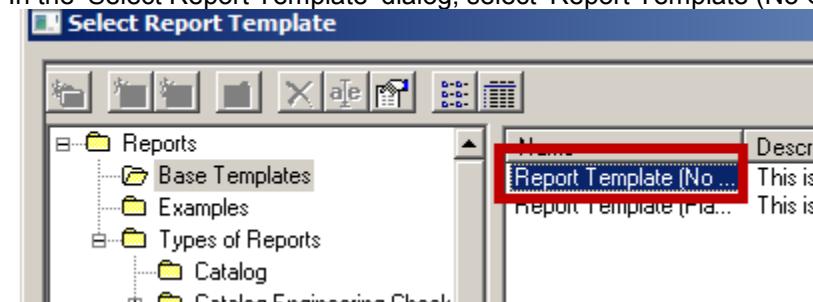


```
SQLQuery5.sql ...ndongre (57)* SQLQuery4.sql ...ndongre (60)* SQLQue
select j2.Oid as 'OID',
j1.ItemName as 'Parent Name',
j2.ItemName as 'Equipment Name'
from XHasEqpAsChild x1
join JNamedItem j1 on j1.Oid = x1.OidOrigin
join JNamedItem j2 on j2.Oid = x1.OidDestination
```

3. Select the query and copy to clipboard
4. Open SmartPlant 3D and switch to the Drawings and Reports task
5. Right mouse click on Spreadsheet Reports and select 'Create Report'

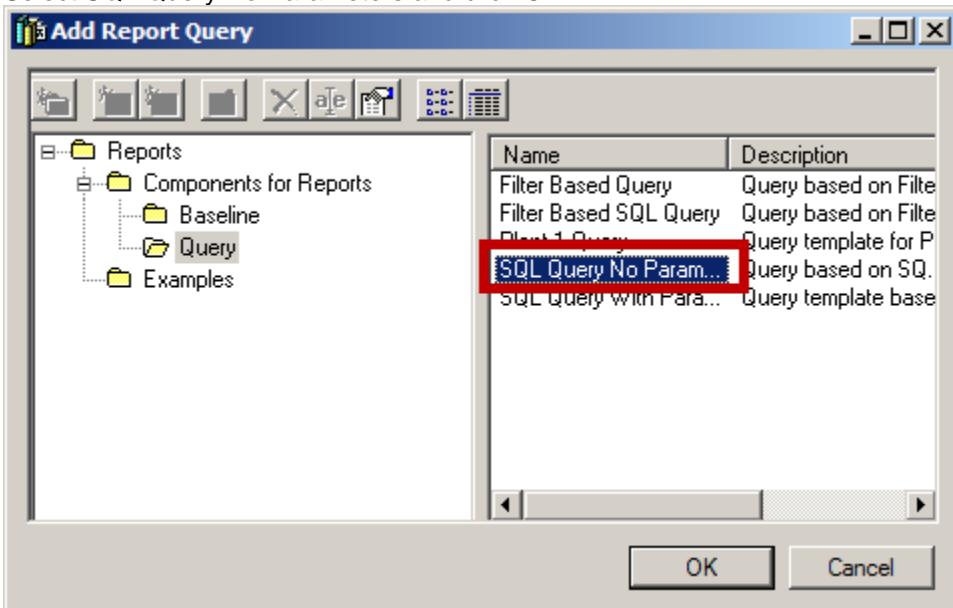


6. In the 'Select Report Template' dialog, select 'Report Template (No Query)' and click OK.

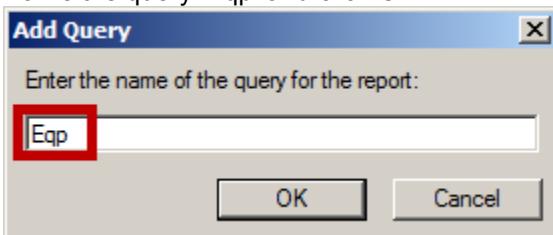


7. Rename the newly created report to 'Equipment Report SQL Query'
8. Right mouse click on the Report Template in the Detail view and select 'Edit Template'
9. Tools → Add Query

10. Select SQL Query No Parameters and click OK



11. Name the query 'Eqp' and click OK.



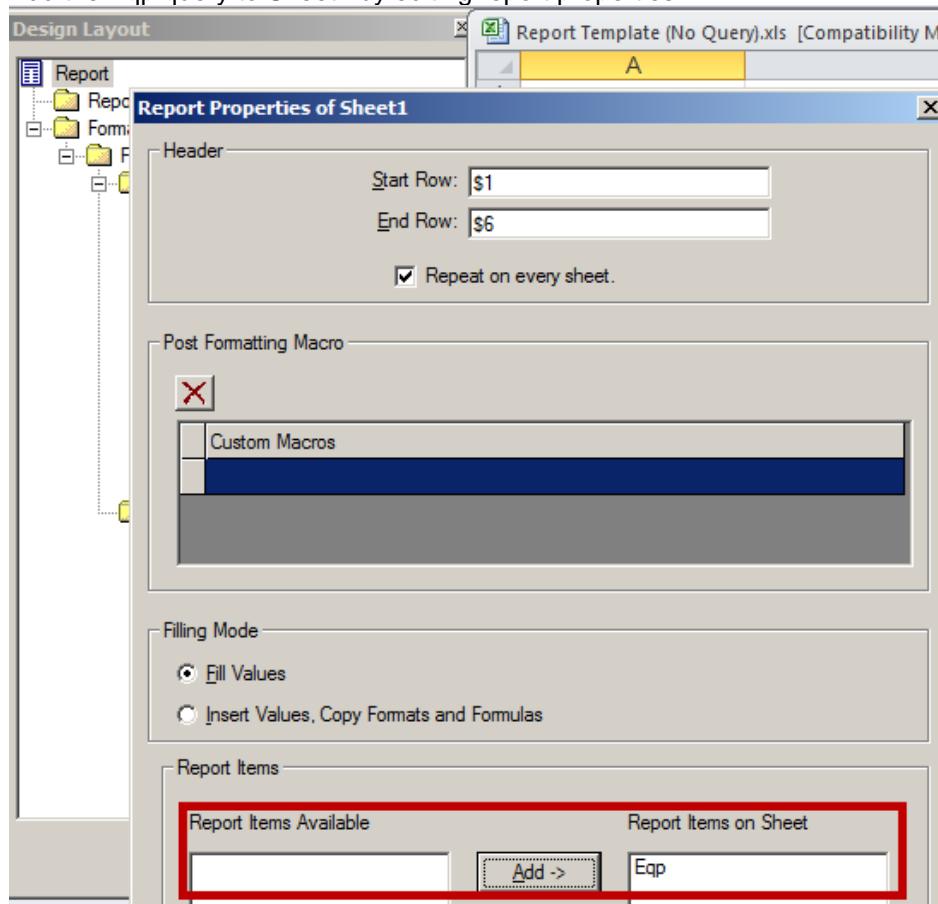
12. Paste the query from the clipboard replacing the statement '—Insert your query statement here'

13. Execute the query using the green triangle inside Smart 3D to verify the query works

OID	Parent Name	Equipment Name
{00004E2E-0000-0000-0000-000000000001}	Equipment	T-101
{00004E2E-0000-0000-0000-000000000002}	Equipment	P-101
{00004E2E-0000-0000-0000-000000000003}	Equipment	B-101
{00004E2E-0000-0000-0000-000000000004}	CT	LG-1

14. Click the Formatting tab and enter Design Layout.

15. Add the Eqp query to Sheet1 by editing report properties.



16. Enter the header information and add the OID, Parent Name and Equipment Name to the cells C9, D9 and E9 as shown

	C	D	E
1			
2			
3			
4			
5			
6			
7	OID	Parent Name	Equipment Name
8	#Eqp::OID#	#Eqp::Parent Name#	#Eqp::Equipment Name#
9			
10			

17. Close Excel and Save when prompted.

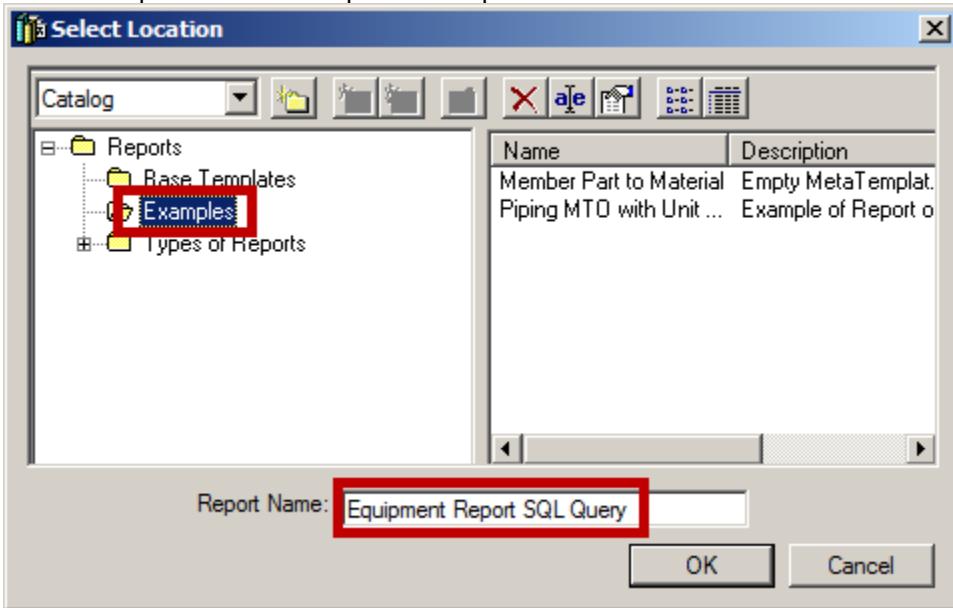
18. File → Save Report Template

19. Close report template

20. Update the report

21. Right click the report and Copy to Catalog

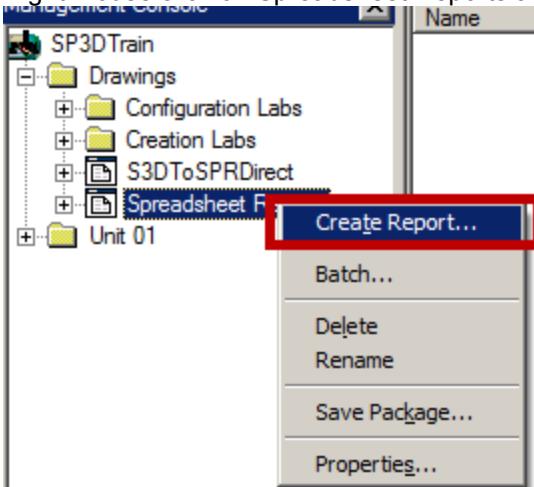
22. Save the report under the Reports\Examples folder



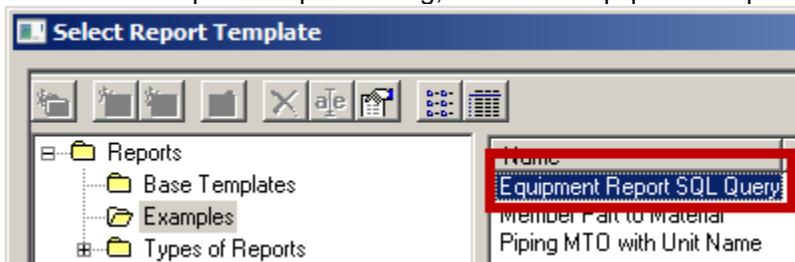
Filter Based SQL Query

The SQL query based report above is not parametric and always runs on the entire model. In order to parameterize the report, it is easiest to turn the query into a Filter Based SQL query instead of a SQL Query with No Parameters

1. Right mouse click on Spreadsheet Reports and select 'Create Report'

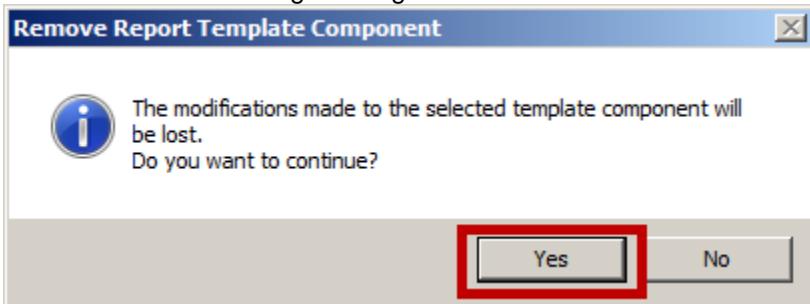


2. In the Select Report Template dialog, select the 'Equipment Report SQL Query' and click OK.

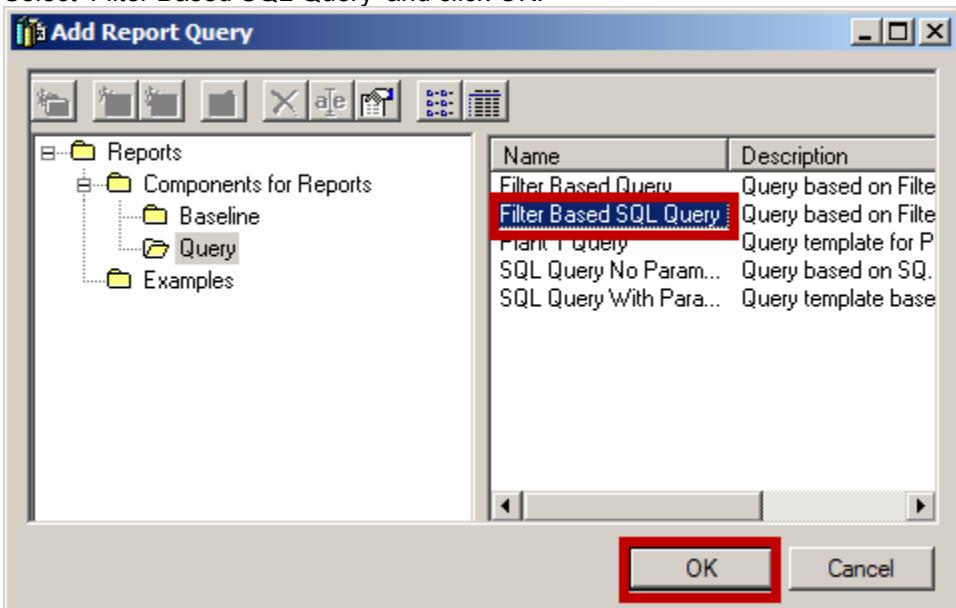


3. Rename the report to 'Equipment Report FBSQL Query' and edit template
4. Select the query statement displayed and copy to clipboard

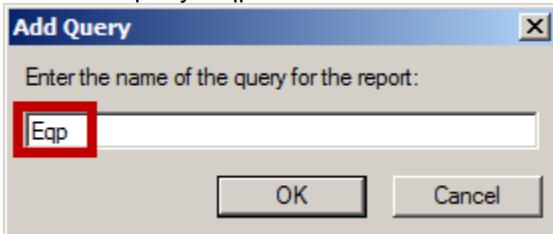
5. Tools → Remove Query:Eqp to remove the SQL query
6. Select Yes to the warning message



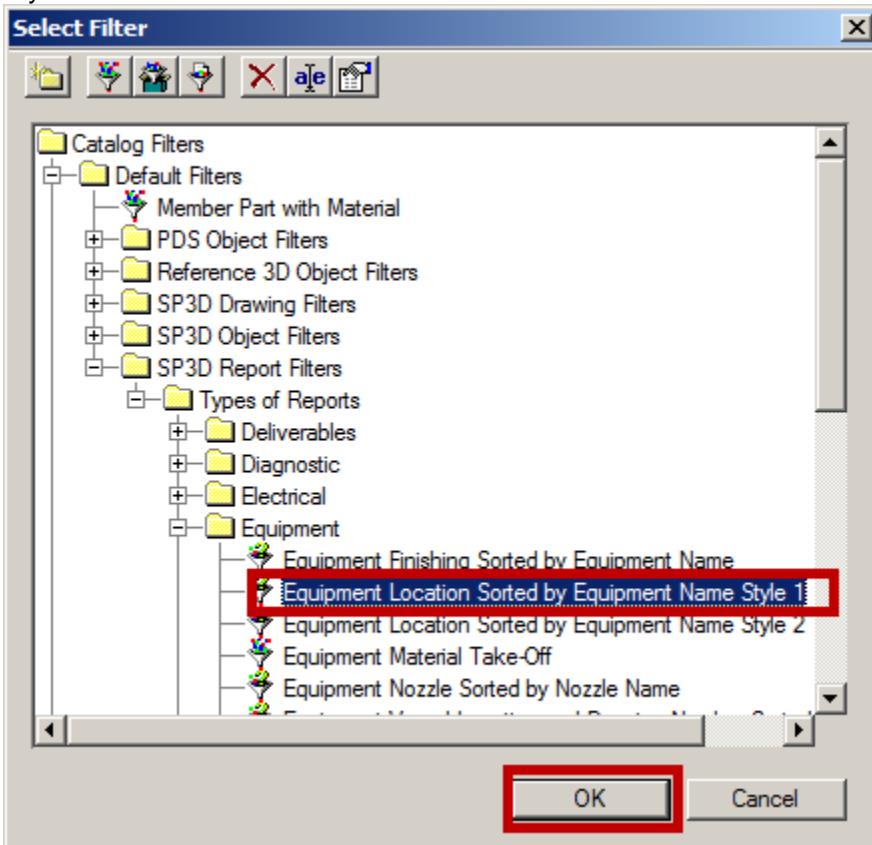
7. Tools → Add Query
8. Select 'Filter Based SQL Query' and click OK.



9. Name the query 'Eqp' and click OK.



10. The Select Filter dialog is shown. Select the 'Equipment Location Sorted by Equipment Name Style 1' filter and click OK.



11. Select the rows as shown below and paste from clipboard

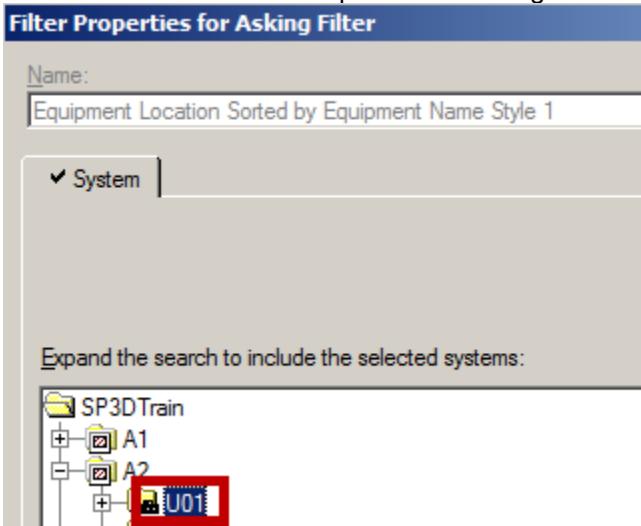
```
/* Insert your query statement here. */
Select ItemName from JNamedItem
WHERE oid IN (SELECT oid FROM @RPTtemptable)
```

12. Edit the WHERE clause to say WHERE j2.oid in

```
select j2.Oid as 'OID',
j1.ItemName as 'Parent Name',
j2.ItemName as 'Equipment Name'
from XHasEqpAsChild x1
join JNamedItem j1 on j1.Oid = x1.OidOrigin
join JNamedItem j2 on j2.Oid = x1.OidDestination
WHERE j2.oid IN (SELECT oid FROM @RPTtemptable)
```

13. Execute query using the green triangle

14. Select 'U01' in the 'Filter Properties for Asking Filter' dialog and click OK.



15. The results pane returns four rows

	OID	Parent Name	Equipment Name
▶	{00004E2E-0000-0000-0000-000000000001}	Equipment	Pump-001
	{00004E2E-0000-0000-0000-000000000002}	Equipment	Pump-002
	{00004E2E-0000-0000-0000-000000000003}	Equipment	E-102
	{00004E2E-0000-0000-0000-000000000004}	Equipment	Electrical Device

16. File → Save Report Template

17. Select U01 and click OK

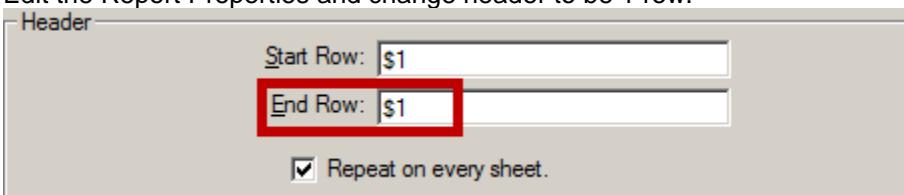
18. Click on the 'Spreadsheet Reports' folder, this closes the report template

19. Update the report.

Enable report to be embedded into a drawing

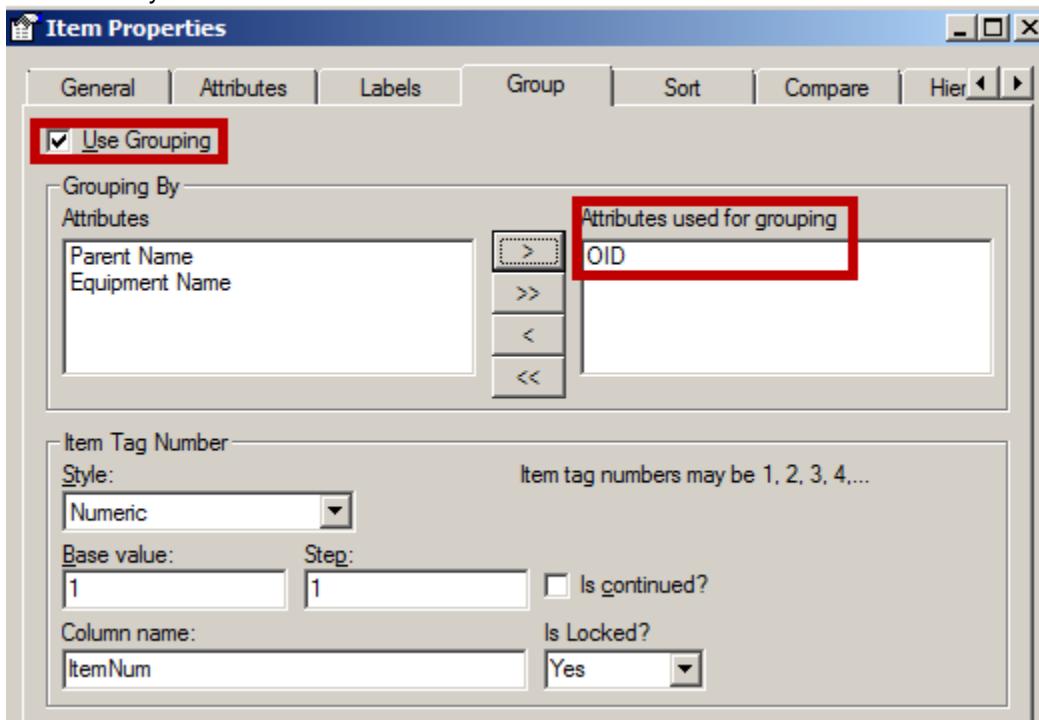
For a report to be embeddable into a drawing, it must be filter based and the OID of a drawable object must be included in a column in the report with the alias OID. Additionally, if bubble-labeling ability is desired, it is required that the report have a column named ItemNum. This is achieved by using grouping functionality. It is also desirable to clean up the header to make it fit better in a drawing view as well as hide the OID column.

1. Edit template on the 'Equipment Report FBSQL Query' report
2. Click on the Formatting tab and enter Design Layout
3. Edit the Report Properties and change header to be 1 row.



4. Right click on the Eqp query and edit properties.
5. Click on the Group tab

- Check the 'Use Grouping' box and group by OID. Notice the ItemNum column created automatically



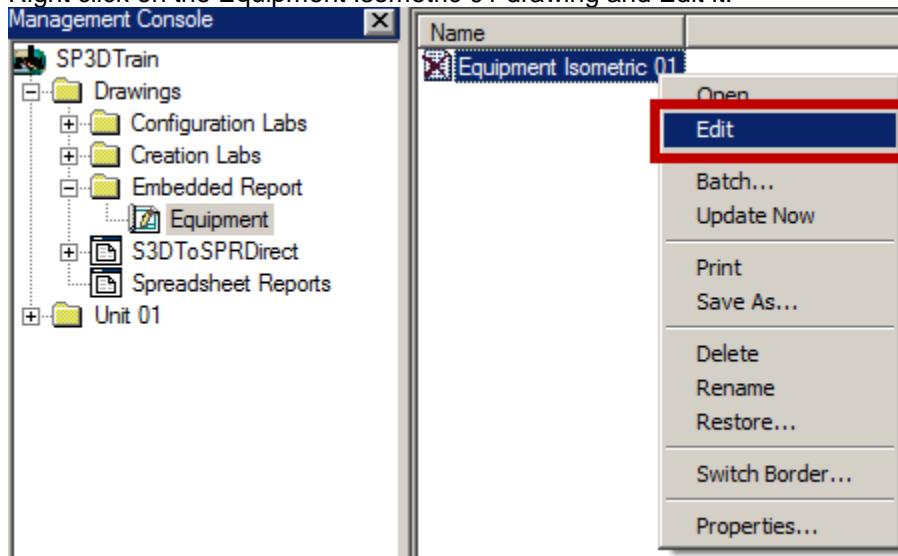
- Click OK
- Select rows 1 through 10 in Excel and delete them
- Enter header in row 1 and drag and drop the fields in row 2 as shown below

A	B	C	D
OID	Item Number	Parent Name	Equipment Name
#Eqp::OID#	#Eqp::ItemNum#Eqp::Parent Name#	#Eqp::Equipment Name#	

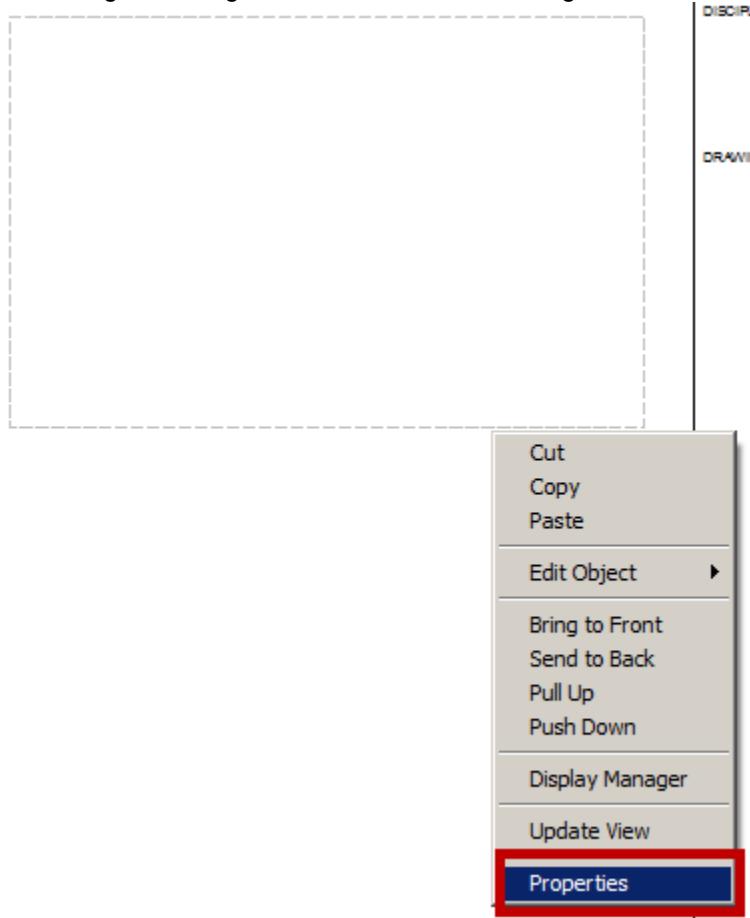
- Select columns E through M and delete them using the right mouse click Delete command. Note that it is not sufficient to press the delete key, this deletes cell contents, we want to delete the cells themselves.
- Select rows 3 through 20 and delete them using the right mouse click Delete command.
- Select cell A1 and press CTRL + SHIFT + END. This should select cells A1 through D2. If any empty cells are selected, select and delete the rows or columns that contain them and save the spreadsheet. After saving, perform the CTRL + SHIFT + END again and check that no empty cells are selected.
- Select column for OID and hide it
- Close Excel and save when prompted.
- File → Save Report Template
- Select U01 and click OK
- Click on the 'Spreadsheet Reports' folder, this closes the report template
- Right click the report and Copy to Catalog
- Save the report under the Reports\Examples folder.

Test Report by Embedding

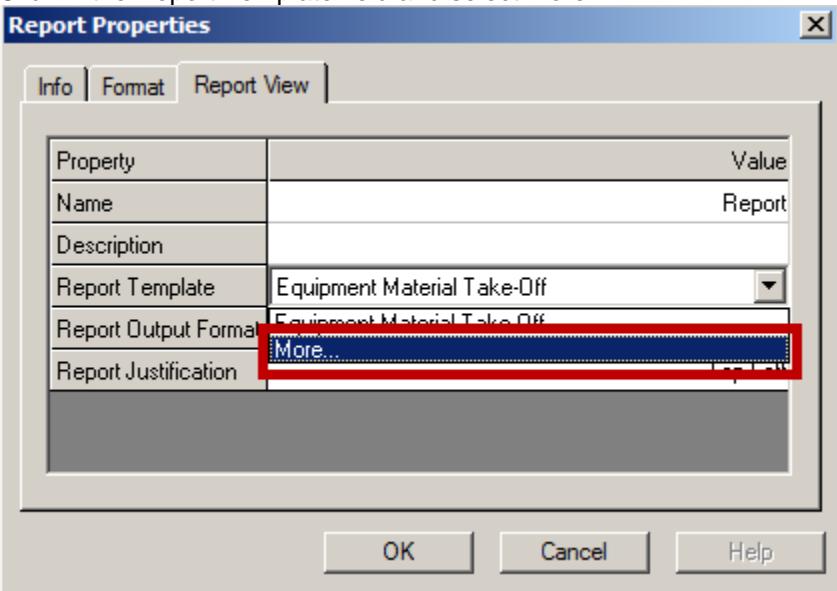
1. Right click on the Equipment Isometric 01 drawing and Edit it.



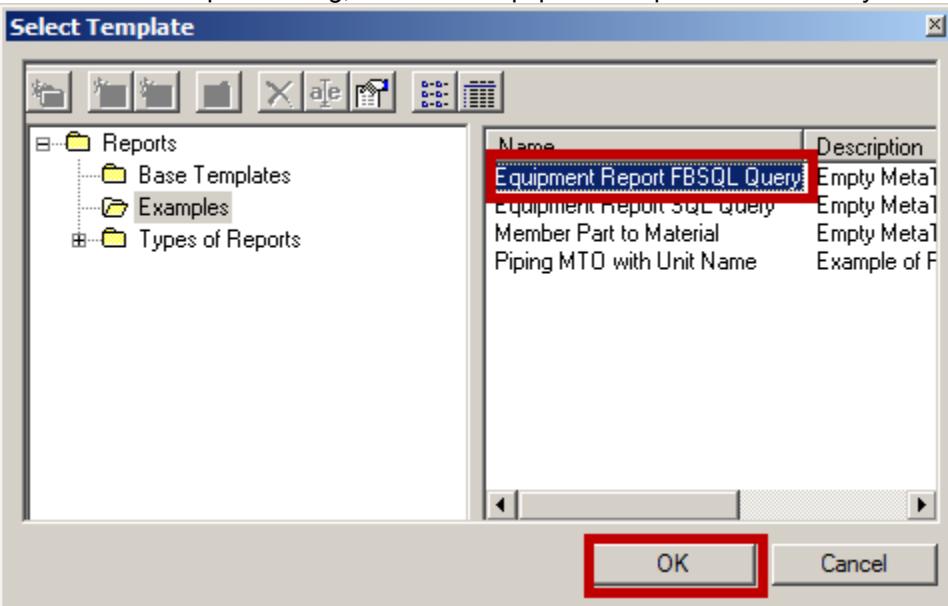
2. In Drawing Editor, right-click on the view to the right and edit Properties



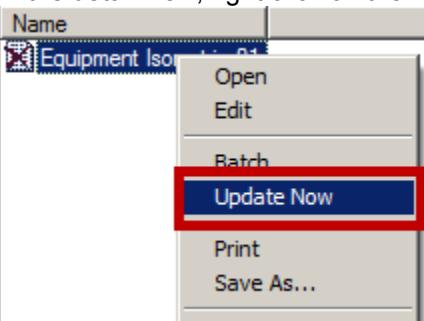
- Click in the Report Template field and select More..



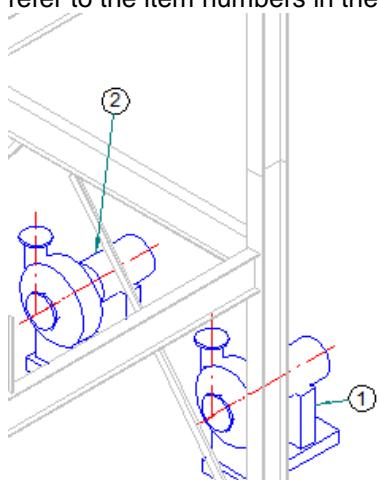
- In the Select Template dialog, select the 'Equipment Report FBSQL Query' and click OK.



- Click OK on the Report Properties dialog.
- Close SmartSketch Drawing Editor
- In the detail view, right click on the 'Equipment Isometric 01' drawing and Update Now.



8. Once update completes, edit the drawing. The equipment in the view contains bubble labels that refer to the item numbers in the report.



Item Number	Parent Name	Equipment Name
1	Equipment	Pump-001
2	Equipment	Pump-002
3	Equipment	Electrical Device

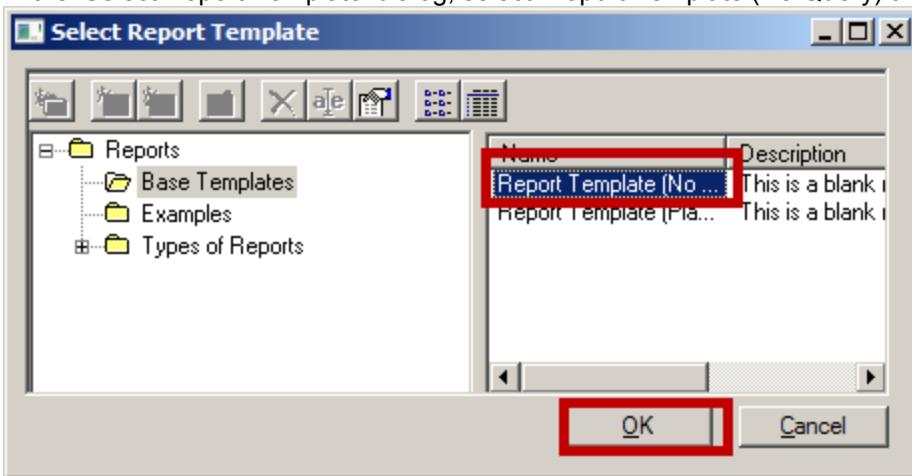
Advanced Reporting

Differential Reporting

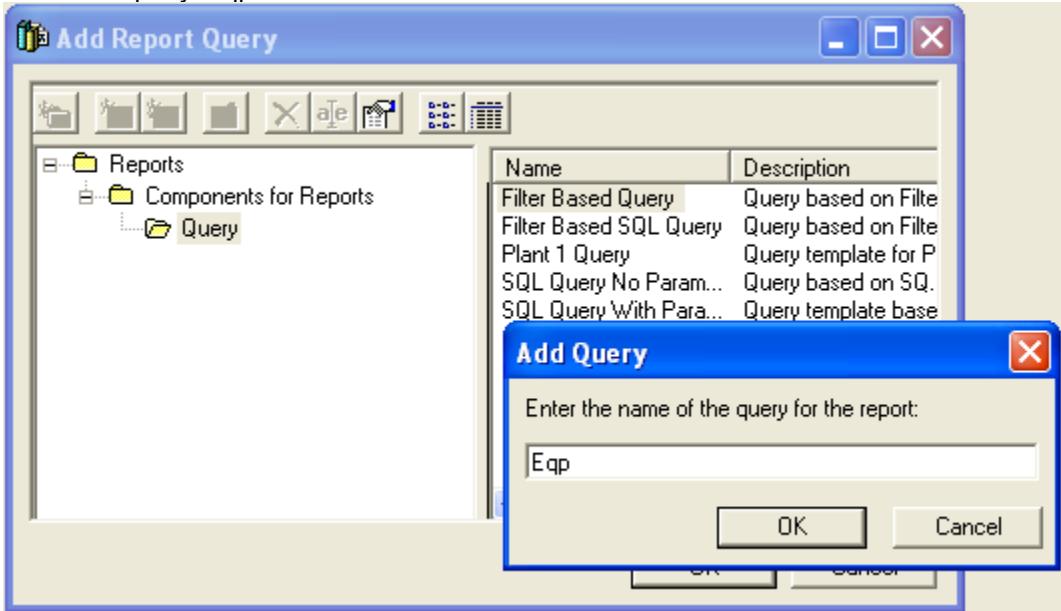
SmartPlant 3D allows users to create a report that will compare against a previously run baseline. Support is available in the user-interface to set up the criteria for comparison.

Creating the report

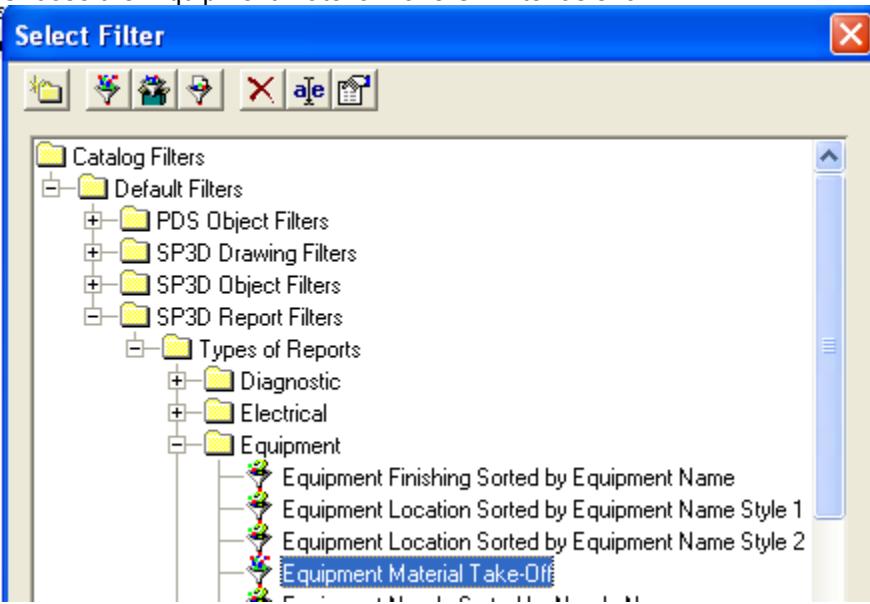
1. Start Smart 3D and define a workspace using the Plant Filters – Training Filters – U04 filter.
2. Switch to the Drawings and Report task.
3. Right mouse click on Spreadsheet Reports and select ‘Create Report’
4. In the ‘Select Report Template’ dialog, select ‘Report Template (No Query) and click OK.



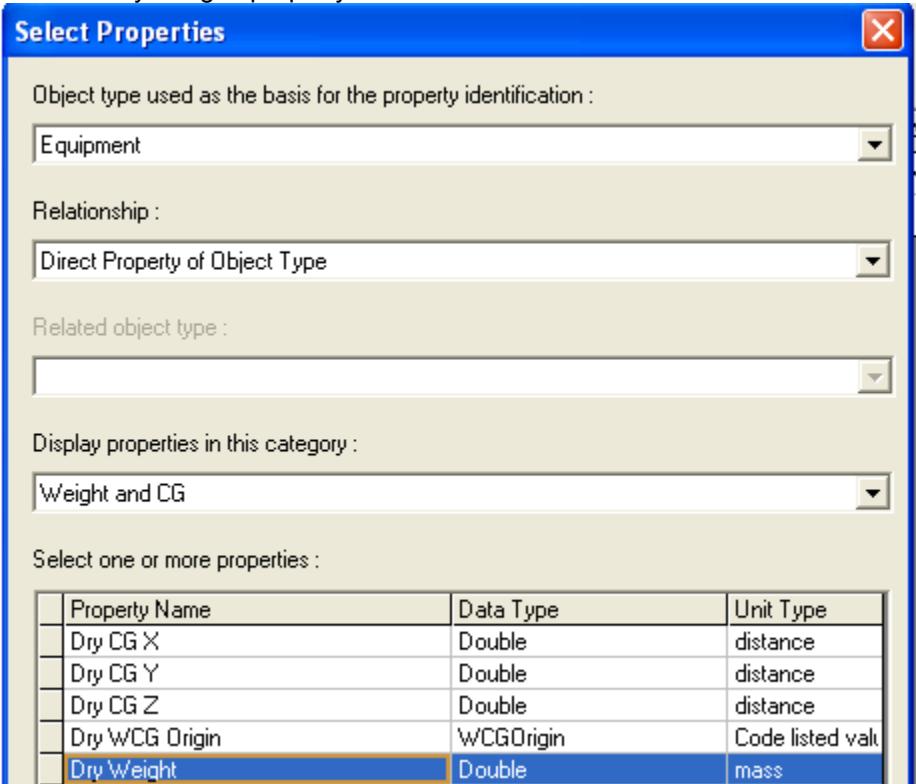
5. Rename the report ‘Equipment Compare’
6. Edit Template and Tools → Add Query
7. Add a Filter Based Query
8. Name the query ‘Eqp’



9. Choose the 'Equipment Material Take Off' filter as shown



10. Add the 'Dry Weight' property

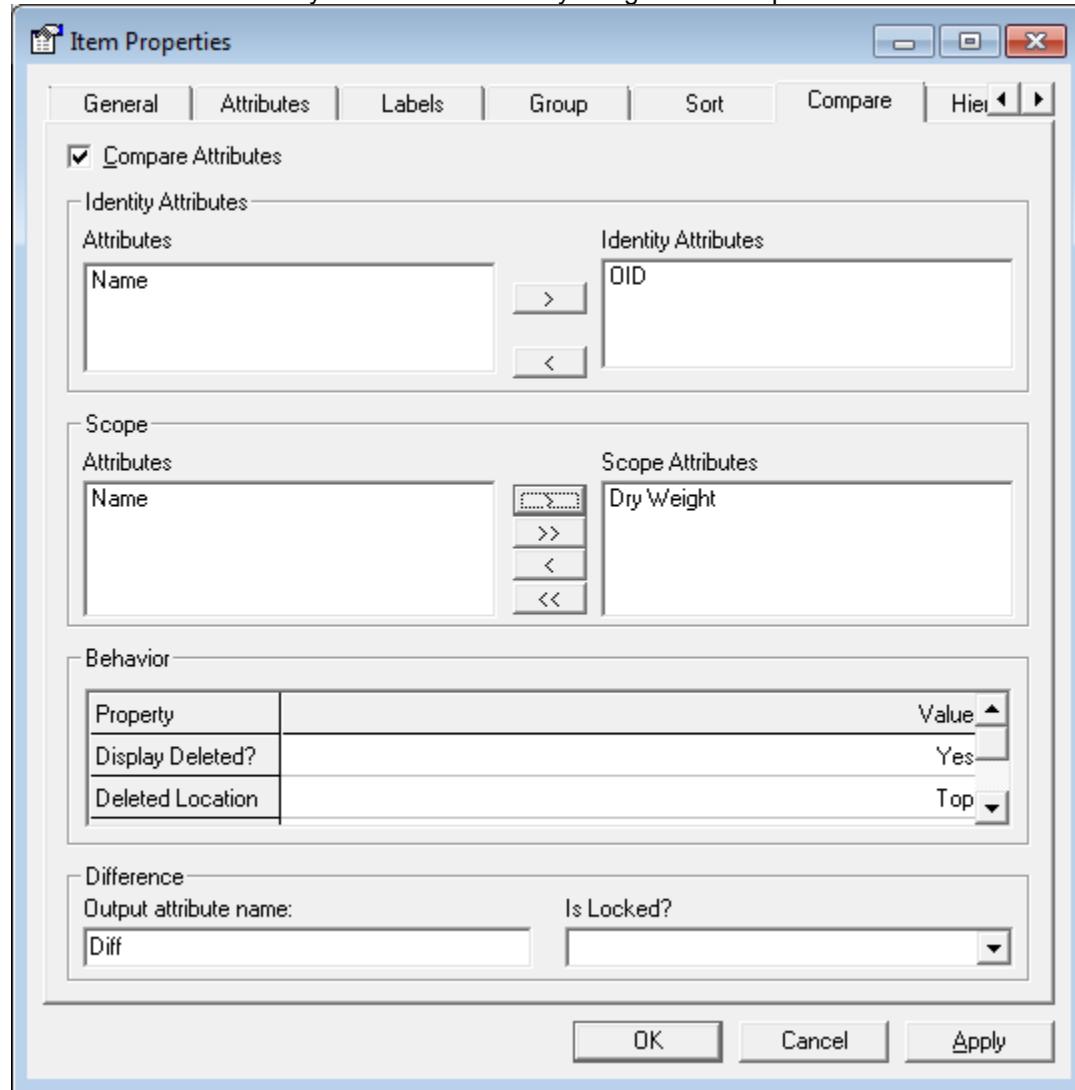


11. Enter Design Layout

12. Add the Eqp query to Sheet1

13. Drag and drop the properties OID, Name and Dry Weight to columns A, B and C

14. Edit properties of the query 'Eqp' and select the Compare tab
 15. Check the box 'Compare Attributes'
 16. Add the OID as an identity attribute and the Dry Weight as a Scope Attribute and click OK.

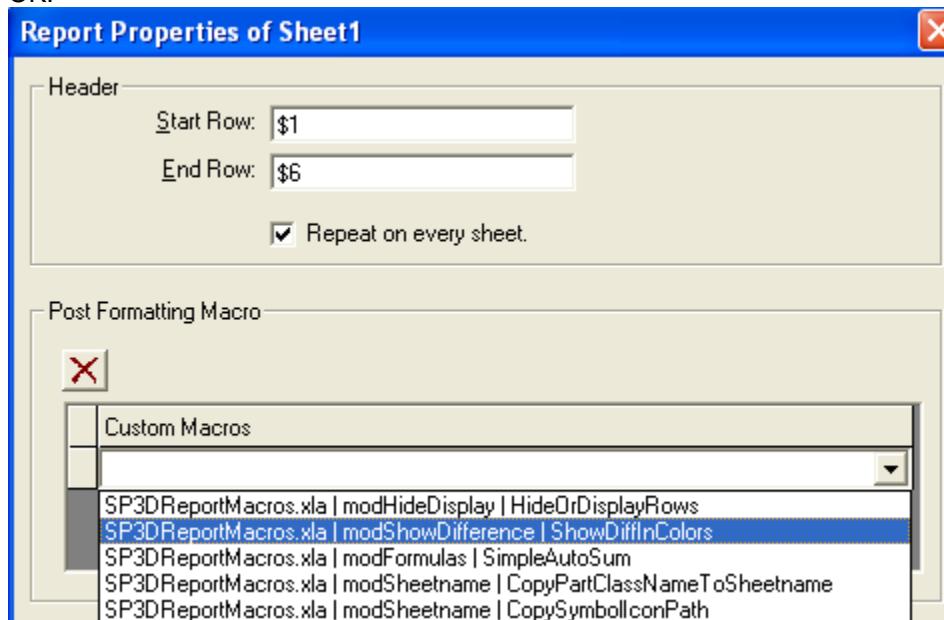


Notes:

- The identity attribute is the basis for comparison of rows in the old and new reports.
- The scope attribute is what decides whether a row appears as modified in the report. Since this example has no grouping, it is best to choose the OID as the identity attribute rather than the name since the name could be the same for two equipment, but the OID is guaranteed to be unique.
- If the dry weight of an equipment changes, it will be highlighted in the report output. If any other property changes, this report will not consider the row as being modified.

17. A new attribute named 'Diff' has appeared. Drag and drop this onto column D

18. Edit Report properties in the design layout tree and add the macro 'ShowDiffInColors' and click OK.



19. Close Excel and click 'Save' when prompted to save.

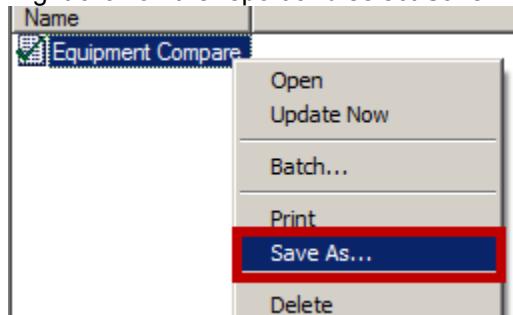
20. Using Tools – Add Baseline, add the report baseline

21. Save Report Template and click Finish on the resulting dialog leaving the baseline prompt empty.

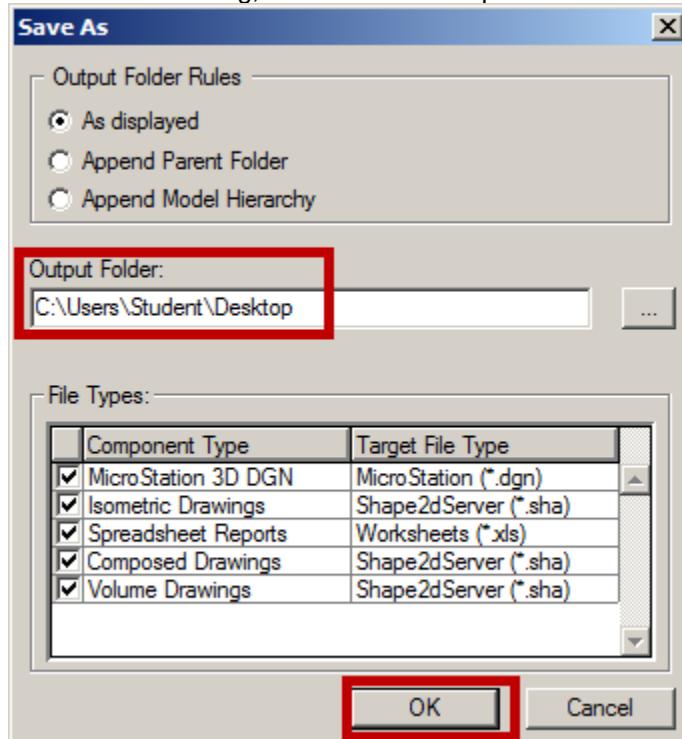
22. File → Save Report Template.
23. Select Finish on the Report Parameters box.
24. Close the report template using the lower close box.
25. Copy Report to Catalog under the 'Examples' folder

Testing the report

1. Update the Report
2. Right click on the report and select Save As...

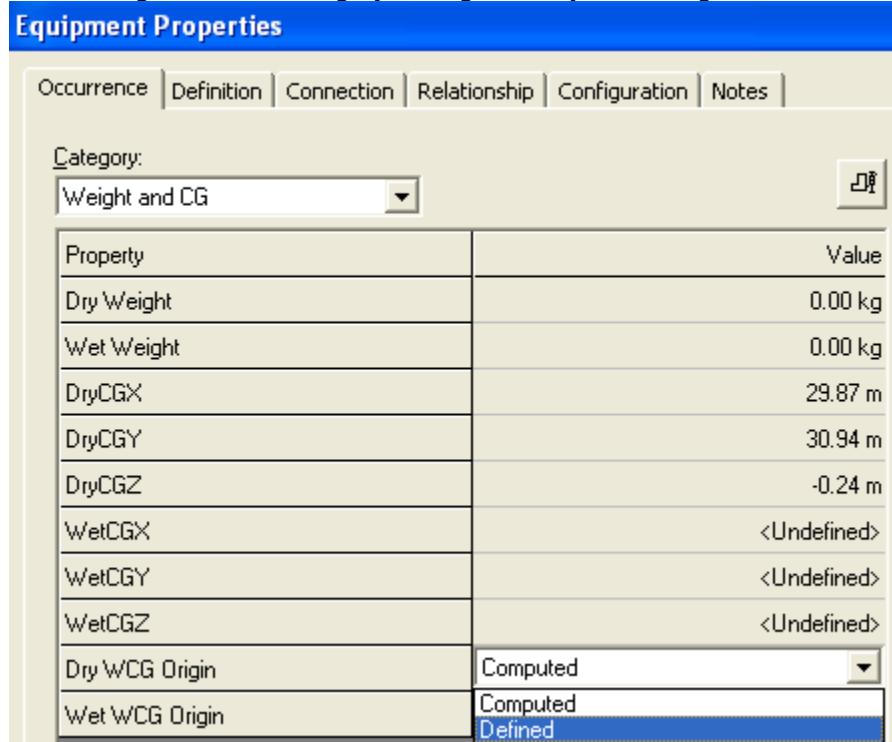


3. In the save as dialog, browser to Desktop and click OK to save the report.

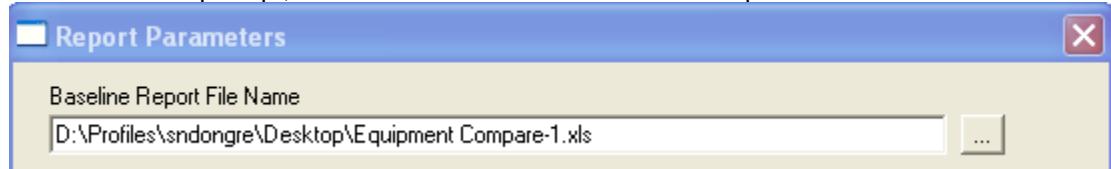


4. Rename the file on desktop to Equipment Compare – 1.xls
5. Switch to 'Equipment' task and edit properties on 40V-101

6. In the 'Weight and CG' category, change the Dry WCG Origin to 'Defined' and click Apply.



7. The Dry Weight field becomes available, enter a value of 100 kg and click OK.
 8. Switch to the 'Drawings and Reports' task.
 9. Right-click the 'Equipment Compare' and select 'Parameters'
 10. At the baseline prompt, browse to the file saved on the desktop and click Finish.

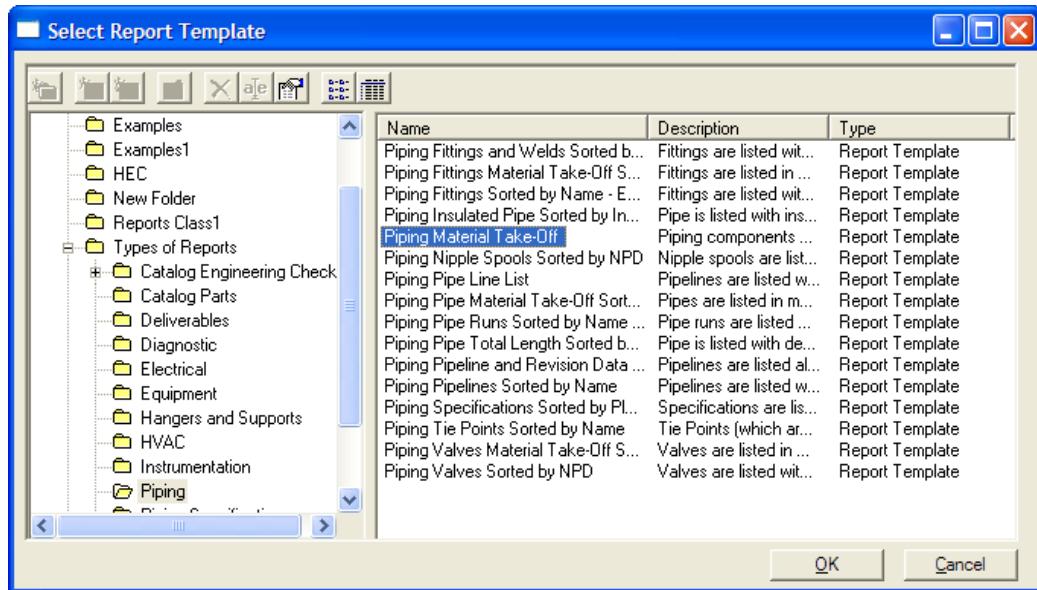


11. Update the report and open it.
 12. Notice that column E shows a 'U' for every row that is unchanged, except the row for 40V-101
 13. There are two rows for 40V-101, a shaded row with an 'O' and a new row with an 'N'
- | 24 | {00004E2E-0000-0000 41P-101A | 740 | U | |
|----|-------------------------------------|-----|---|--|
| 25 | {00004E2E-0000-0000 41P-101B | 740 | U | |
| 26 | {00004E2E-0000-0000 40E-101B | 0 | U | |
| 27 | {00004E2E-0000-0000 40V-101} | 0 | O | |
| 28 | {00004E2E-0000-0000 40V-101 | 100 | N | |
| 29 | {00004E2E-0000-0000 T-101 | 0 | U | |
| 30 | {00004E2E-0000-0000 E-102 | | U | |
14. Similarly you may expect deleted rows shown with a 'D' and new rows with an 'A'

Hierarchical Reporting

We'll modify the delivered Piping Material Take-Off report to make it a report that reports by pipeline.

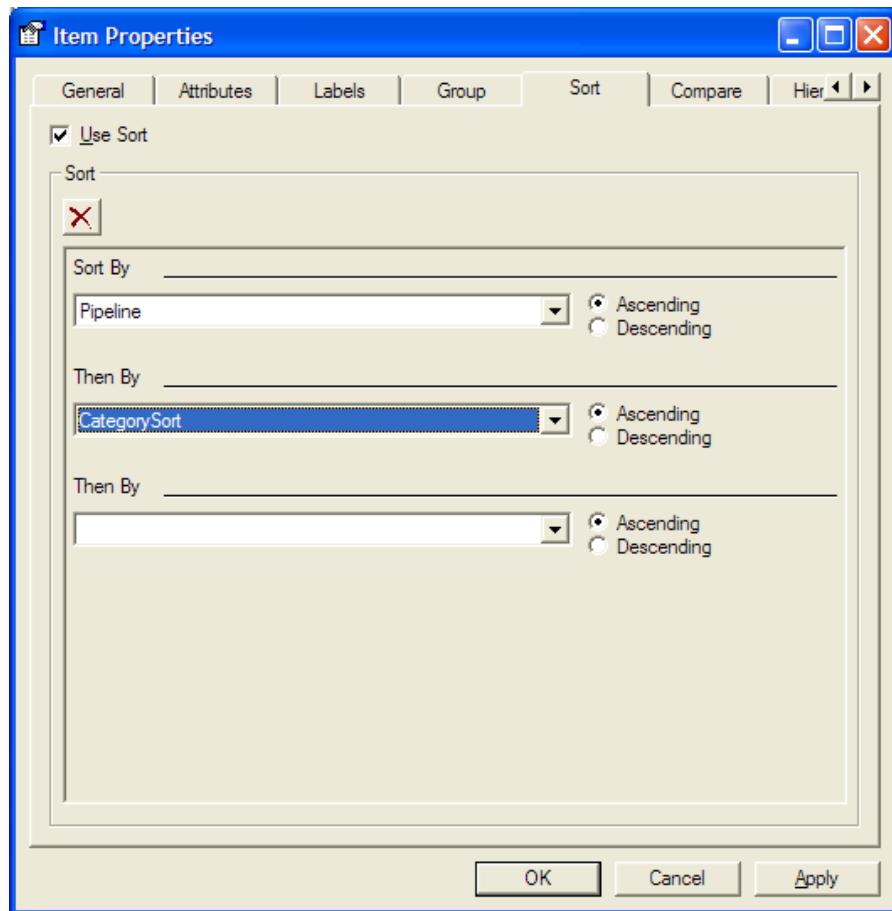
- Create a new report. Instead of starting from an empty template, select Types of Reports\Piping\Piping Material Take-Off.



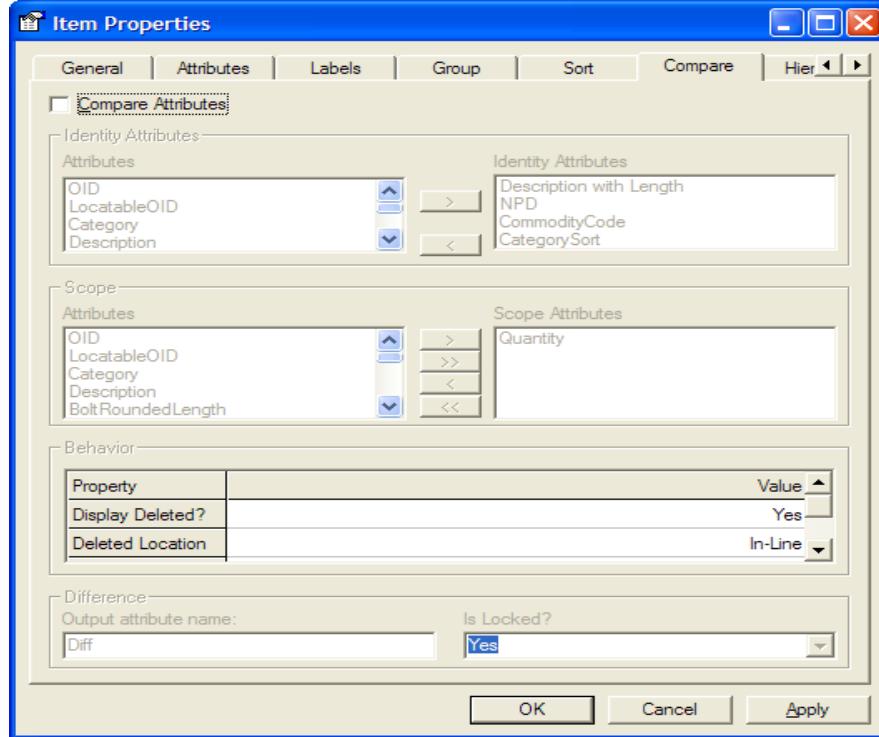
2. Select the A2/U01 node when prompted by the filter dialog. Click “Next” at the baseline prompt and “Finish” at the UOM dialog.
3. Rename the report ‘Piping Material Take-Off Sheet Per Pipeline’
4. Edit Template and go into the Design Layout. Insert a column before the *Item Number* Column (E).
5. Enter ‘Pipeline’ in cell E1 and apply formatting from cell F1 to it using Format Painter command.
6. Drag Pipeline attribute to cell E3

E	F	G	H
1			
2			
3	#Component::Pipeline#	#Component::ItemNum#	#Component::Category#
4			
5			

7. Right-click *Component/Properties*.
8. Under the *Group* tab add *Pipeline* to the Attributes Used for Grouping.
9. Under the *Sort* tab add “*Pipeline*” as the first attribute to sort by and “*Category Sort*” as the second attribute to sort by.

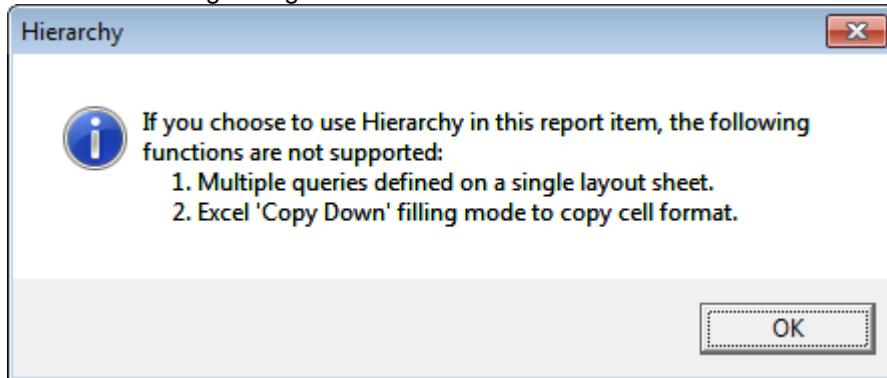


10. Remove the check box on the *Compare* tab

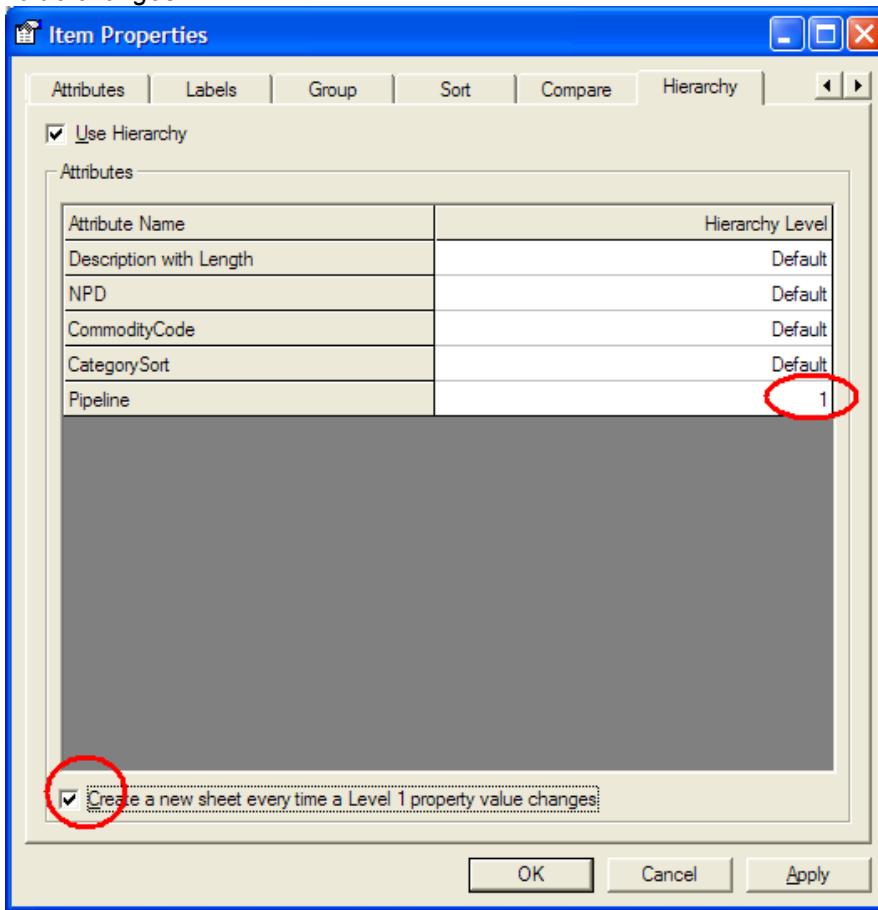


11. Click OK to save settings

12. Edit query properties again and check the check box “*Use Hierarchy*” on the *Hierarchy* tab. Click OK on the warning dialog:



13. Set Pipeline as a Level 1 attribute, and check “*Create new sheet every time a Level 1 property value changes*”



Note:

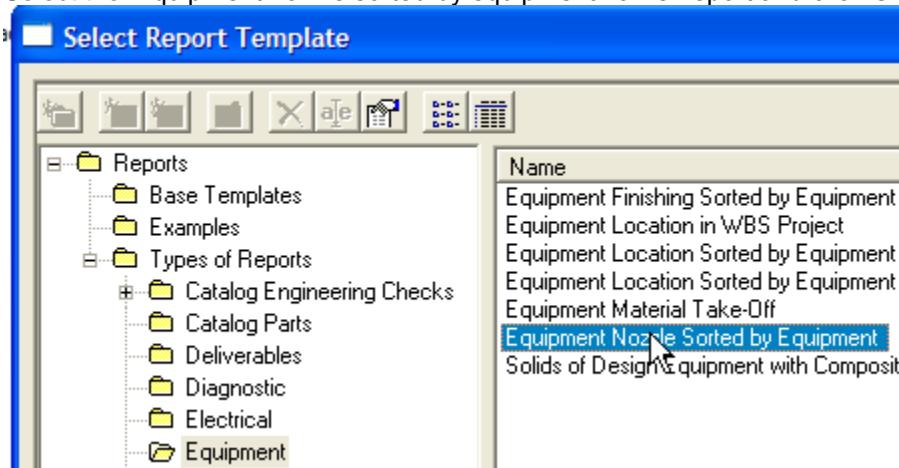
- When using Hierarchy, the value for a cell in a given row and a given column will be skipped (not output) according to the rules:
 - If that column attribute is Hierarchy level *Default* – never.
 - If that column attribute is Hierarchy level *k* – when all the values in the present row for columns of hierarchy levels between 1 and *k* including are the same as the values for those same columns in the preceding row.

14. Click OK to save settings. This will create and output the information for each pipeline on a separate sheet, and that pipeline name will be output only once.
15. Close excel, save when prompted.
16. Save report template.
17. Update report. You should see two sheets with each sheet containing information for one pipeline.

Delegated Query

Adding connected run name to nozzle report

1. In Drawings and Reports task, right mouse click on 'Spreadsheet Reports' and select 'Create Report'
2. Select the 'Equipment nozzle sorted by equipment name' report and click OK.

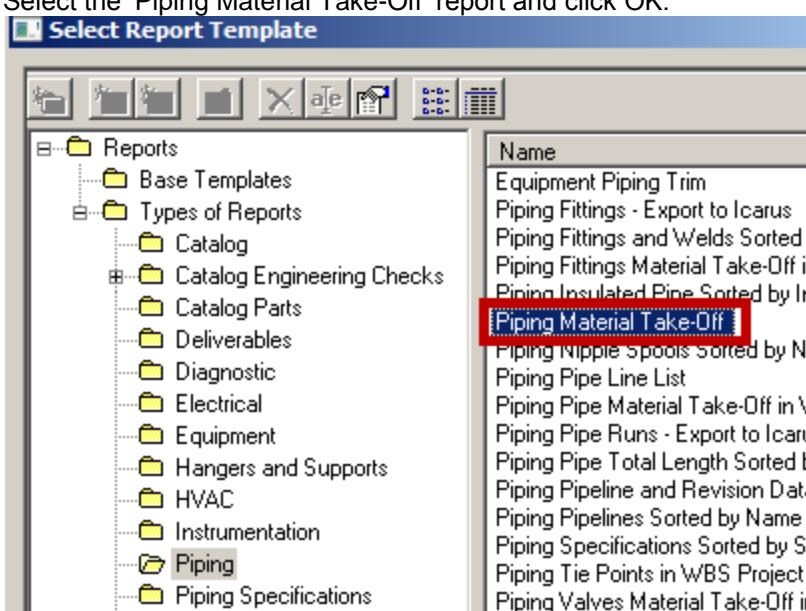


3. Select A2 - U01 system and click Next and Finish on the prompts.
4. Rename the newly added report 'Equipment Nozzle with Connected Runs'
5. Edit Template on the new report
6. Save report template
7. Select A2 - U01 system and click Next and Finish on the prompts.
8. Copy to Catalog the report under the 'Examples' folder
9. Navigate to the location on symbol share where the files are stored and open the rqe file.
(Note: There are two rqe files, one containing a SQL query for plant, and the second a COM query containing the returned properties. The query file to open is the one containing the COM query.)
10. Open the rqe file for the 'Nozzle to Connected Run – Using Edge'.
11. Select the <RETURNED_PROPERTY> to </RETURNED_PROPERTY> from the label rqe and copy it into the report's rqe file just below the existing </RETURNED_PROPERTY>
12. Save the report rqe file
13. In Drawings and Reports task, right mouse click on 'Spreadsheet Reports' and select 'Create Report'
14. Select the 'Equipment nozzle with Connected Runs' report and click OK.
15. Select A2 - U01 system and click Next and Finish on the prompts.
16. Edit template on the new report
17. Click on the formatting tab and click 'Design Layout'
18. Drag the new property 'Run Name' to column 'L'
19. Close Excel and save when prompted.
20. Save report template
21. Select A2 - U01 system and click Next and Finish on the prompts.

22. Update the report.

Adding unit name label to piping MTO

1. In Drawings and Reports task, right mouse click on 'Spreadsheet Reports' and select 'Create Report'
2. Select the 'Piping Material Take-Off' report and click OK.

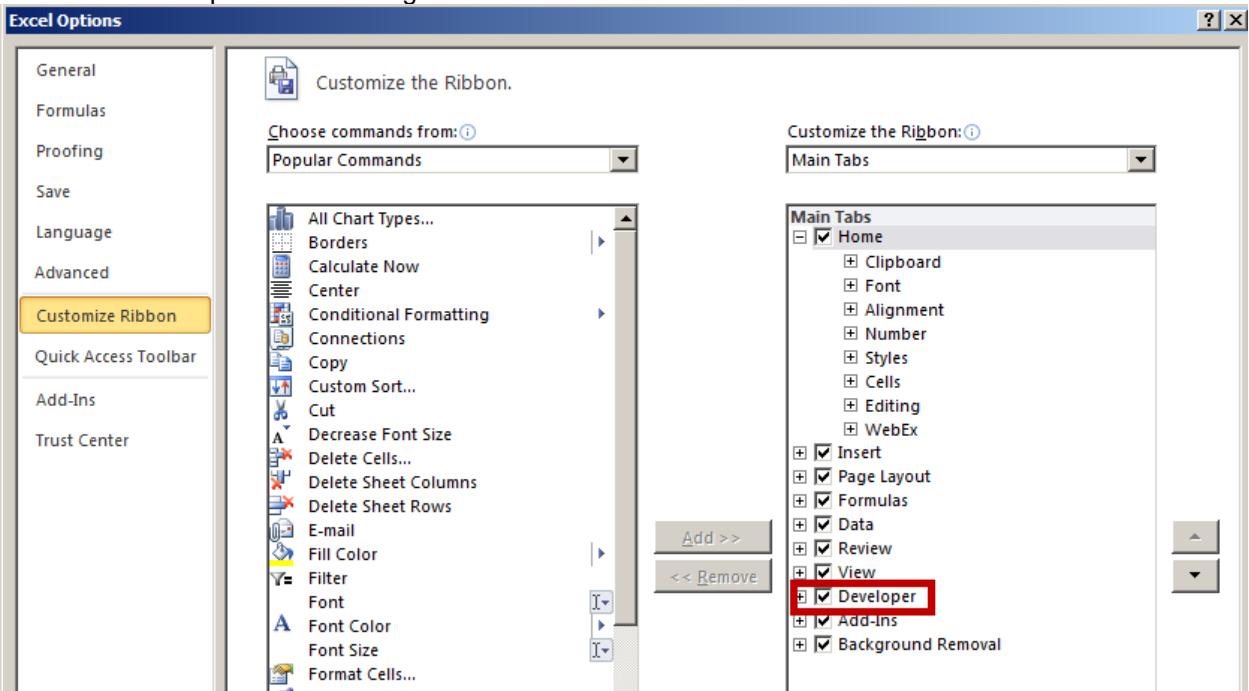


3. Select A3 system and click Next and Finish on the prompts.
4. Rename the newly added report 'Piping MTO With Unit Name'
5. Edit Template on the new report
6. File → Save Report Template
7. Copy to Catalog the report under the 'Examples' folder
8. Navigate to the location on symbol share where the files are stored and open the rqe file
9. Open the rqe file for the 'Unit Name' label defined earlier.
10. Select the <RETURNED_PROPERTIES> to </RETURNED_PROPERTIES> from the label rqe into the report's rqe file just below the existing </ORASQL>
11. Save the report rqe file
12. In Drawings and Reports task, right mouse click on 'Spreadsheet Reports' and select 'Create Report'
13. Select the 'Piping MTO with Unit Name' report and click OK.
14. Select A3 system and click Next and Finish on the prompts.
15. Edit template on the new report 'Piping MTO with Unit Name-1-0001'
16. Click on the formatting tab and click 'Design Layout'
17. Drag the new property 'Unit Name' to cell 'N3'
18. Close Excel and save when prompted.
19. File → Save report template
20. Update the report.
21. Open the completed report to see that unit names U13, U14 and U15 appear in column N. Note that for accuracy of the report, it will be necessary to include the unit name in the grouping criteria as well.

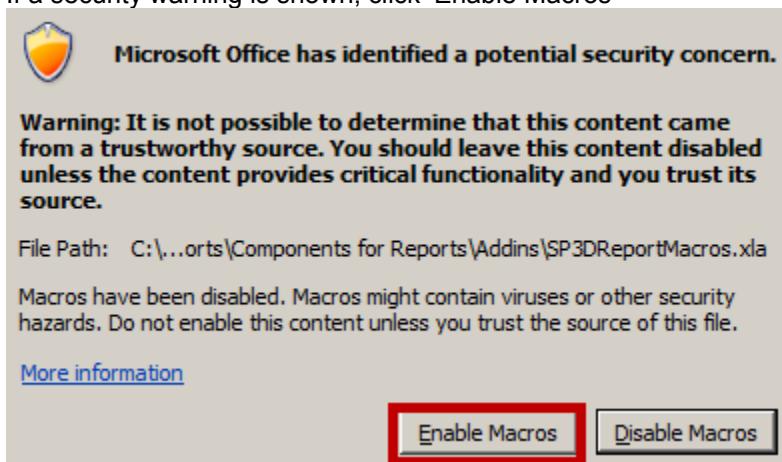
Modifying Macros

Copy pipeline name to sheet name in multi-sheet report

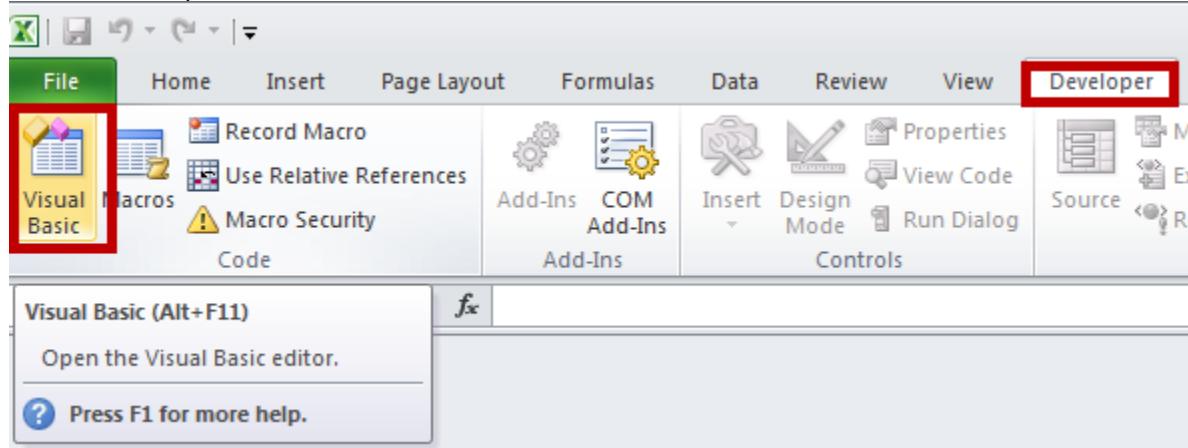
1. Start Microsoft Excel
2. If using Excel 2010, go to File → Options
3. Click Customize Ribbon in the Excel Options dialog
4. Check the Developer box on the right side



5. Click OK on the Excel options form and close Excel.
6. Open SP3DReportMacros.xla in SharedContent\Reports\Components for Reports\Addins
7. If a security warning is shown, click 'Enable Macros'



8. On the 'Developer' tab, click 'Visual Basic'



9. Double-click on the modSheetName module
10. Copy the rows for the sub CopyPartClassNameToSheetName() and paste them right below itself

```

Option Explicit
Public m_PartClassName As String
Public m_SymbolIconPath As String

Public Sub CopyPartClassNameToSheetname()
    Dim oDataSheet As Worksheet
    Dim sText As String
    On Error GoTo ErrorHandler

    For Each oDataSheet In Application.Worksheets
        If oDataSheet.Visible = xlSheetVisible Then
            If oDataSheet.Name <> "SP3DReport_Layout" And _
oDataSheet.Name <> "SP3DReport_Defini" And _
oDataSheet.Name <> "Index" And _
oDataSheet.Name <> "CustomInterfaces" And _
oDataSheet.Name <> "GUIDs" Then
                With oDataSheet
                    sText = .Cells(6, "A") 'where the
                    m_PartClassName = sText
                    .Name = sText
                    .Cells(6, "A") = "" 'if you do
                End With
            End If
        Next
        GoTo Shutdown
ErrorHandler:
    Shutdown:
End Sub

```

11. Change the name of the copied Sub to 'CopyPipelineNameToSheetname'

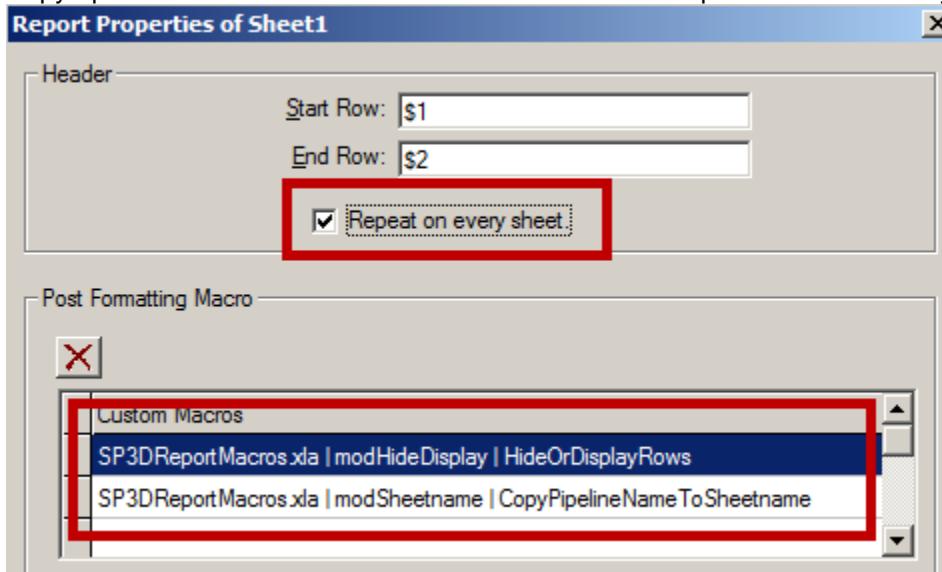
12. To name each sheet with the name of the corresponding pipeline, edit Line 14 of the Sub 'CopyPipelineNameToSheetname' to be cell 3, E (to match cell in the hierarchical report). Also, comment out line 17 that sets cell 6A to blank and save the xla file. Your code should look like below

```
Public Sub CopyPipelineNameToSheetname()
    Dim oDataSheet As Worksheet
    Dim sText As String
    On Error GoTo ErrorHandler

    For Each oDataSheet In Application.Worksheets
        If oDataSheet.Visible = xlSheetVisible Then
            If oDataSheet.Name <> "SP3DReport_Layout" And
                oDataSheet.Name <> "SP3DReport_Definition" And
                oDataSheet.Name <> "Index" And
                oDataSheet.Name <> "CustomInterfaces" And
                oDataSheet.Name <> "GUIDs" Then
                With oDataSheet
                    sText = .Cells(3, "E") 'where the part class name is
                    m_PartClassName = sText
                    .Name = sText
                    '.Cells(6, "A") = "" 'if you do not want to change the sheet name
                End With
            End If
        End If
    Next
    GoTo ShutDown
ErrorHandler:
|
ShutDown:
End Sub
```

13. Save the module
14. Debug → Compile modHideDisplayModule
15. If you have an existing session, close and re-enter Smart 3D and the Drawings and Reports task.
16. Edit the 'Piping Material Take-Off Sheet Per Pipeline' report

17. On Report Properties remove the Macro ShowDiffInColors . Add the macro CopyPipelineNameToSheetname. Also check the box to repeat header on every sheet.



18. Update the 'Piping Material Take-Off Sheet Per Pipeline' report. The report has been split by sheet and each sheet is named after the pipeline it is reporting on.

	E	F	G	I	J
1	Pipeline	Item Number	Category	Description	NPD
3	1001-P	1	Piping	Pipe, S-STD, BE, ASTM-A53-B Type S	10 in
9		2	Fittings	45 deg LR elbow, S-STD, BE, ASTM-A234-WPB, ASME-	10 in
10		3	Fittings	90 deg LR elbow, S-STD, BE, ASTM-A234-WPB, ASME-	10 in
11		4	Fittings	Concentric reducer, S-STD x S-STD bore, BE, ASTM-A2:	10 in x 8
13		5	Fittings	Tee, S-STD, BE, ASTM-A234-WPB, ASME-B16.9	10 in
14		6	Flanges	Flange, CL150, RFFE/BE, ASTM-A105, ASME-B16.5, W	10 in
18		7	Flanges	Flange, CL150, RFFE/BE, ASTM-A105, ASME-B16.5, W	8 in
20		8	Valves	Check valve, CL150, RFFE, BC, swing, ASTM-A216-WC1	10 in
22		9	Valves	Gate valve, CL150, RFFE, BB, OS&Y, ASTM-A216-WCB	10 in
24		10	Bolts	Studbolts, ASTM-A193-B7	4.50 in
26		11	Bolts	Studbolts, ASTM-A193-B7	4.75 in
32		12	Gaskets	Gasket, CL150, 0.125" thk, 304 spiral wnd, graph filled,	C 10 in
38		13	Gaskets	Gasket, CL150, 0.125" thk, 304 spiral wnd, graph filled,	C 8 in
40					

At the bottom of the table, there are navigation buttons: back, forward, search, and other report-related icons. The text '1001-P' is highlighted in red.

Bulkloading reports and labels to catalog

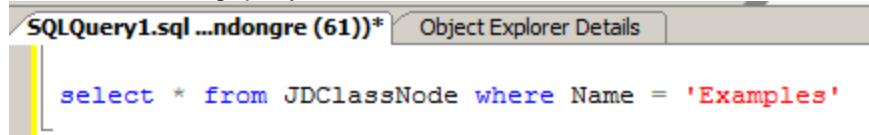
Labels and reports can be added to the catalog one by one using the Tools → Define Label command in catalog task and the Copy to Catalog command in the Drawing and Reports task. However when large numbers of labels need to be added to a catalog, this method is not efficient. Manual creation has the added disadvantage of creating different internal and user names for labels and reports which hurts portability.

Finding out internal name of manually created labels and reports

By running queries on the following five views in the catalog database it is possible to obtain all the information necessary to find internal names of manually created labels and reports

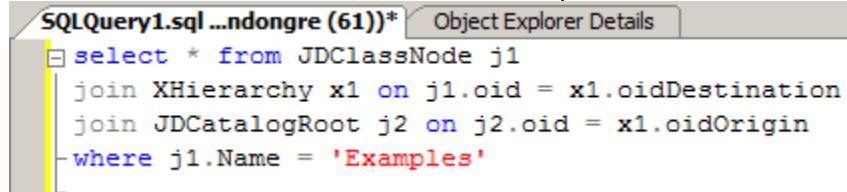
JDCatalogRoot – contains the oids and usernames of catalog roots including labels and reports roots
JReport – contains information about the report and label templates
JDClassNode – contains information about label folders and report folders
XHierarchy – contains the relationships between root, folder and label/report
NamedObjectsView – contains the internal names of all objects

1. Start SQL Server Management Studio and make a connection to your server
2. Select your catalog database _CDB and start a new query
3. Enter the following query and execute



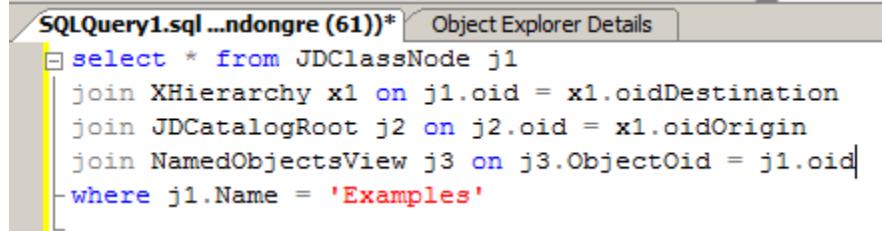
```
SQLQuery1.sql ...ndongre (61)* Object Explorer Details
select * from JDClassNode where Name = 'Examples'
```

4. This returns two rows, one is a labels folder and another is a reports folder. To find out which one is which, we need to execute some additional queries. Enter the following query and execute



```
SQLQuery1.sql ...ndongre (61)* Object Explorer Details
select * from JDClassNode j1
join XHierarchy x1 on j1.oid = x1.oidDestination
join JDCatalogRoot j2 on j2.oid = x1.oidOrigin
where j1.Name = 'Examples'
```

5. This returns additional columns named Labels and Reports, and tells us which object is which. To find the internal names of these, we must make another join with the NamedObjectsView. Enter the following query and execute



```
SQLQuery1.sql ...ndongre (61)* Object Explorer Details
select * from JDClassNode j1
join XHierarchy x1 on j1.oid = x1.oidDestination
join JDCatalogRoot j2 on j2.oid = x1.oidOrigin
join NamedObjectsView j3 on j3.ObjectOid = j1.oid
where j1.Name = 'Examples'
```

6. This query returns an additional column ObjectName where the names are of the type ClassNode-1-xxx. Find the name that corresponds to the Examples folder under the Labels root and note it down for the next step.

Loading additional label content from eCustomer

The instructor will provide the location for the Additional Labels Content.zip file (or you may download it from eCustomer).

There are three sheets in the bulkload workbook that are relevant to loading custom labels and reports.

The Reports sheet contains the actual label or report name, type, description and path relative to the SharedContent folder.

The ClassNodeType sheet is where folders (in the labels and reports hierarchy) are created. Folders have an internal name (ObjectName) and a user name (Name)

The R-Hierarchy sheet establishes the parent-child relationships between the labels or reports root and folders, folders and other folders and labels.

1. Open the Additional Labels Content V1.xls file.
2. On the ClassNodeType sheet, replace LblExamples with ClassNode-1-xxx noted in previous step.

	A	B	C
1	Head	<u>ObjectName</u>	<u>Name</u>
2	Start		
3	a	ClassNode-1-xxx	Examples
4	End		
5			
6			

Report ClassNodeType R-Hierarchy

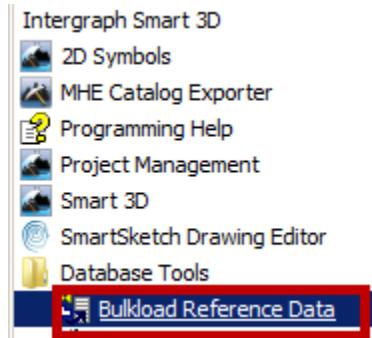
3. On the R-Hierarchy sheet, replace LblExamples with ClassNode-1-xxx noted in previous step.

Head	<u>RelationSource</u>	<u>RelationDestination</u>
Start		
a	RefDataLabelsRoot	ClassNode-1-xxx
a	ClassNode-1-xxx	Area and Unit
a	ClassNode-1-xxx	Cable Path
a	ClassNode-1-xxx	Equipment Bearing
a	ClassNode-1-xxx	Equipment Pitch
a	ClassNode-1-xxx	Equipment Roll
a	ClassNode-1-xxx	Feature Length
a	ClassNode-1-xxx	Iso Run Name
a	ClassNode-1-xxx	Opening Position
a	ClassNode-1-xxx	Pipe Turn Feature Label
a	ClassNode-1-xxx	R3D File Name
a	ClassNode-1-xxx	Run Length
a	ClassNode-1-xxx	Support Notes
a	ClassNode-1-xxx	Supporting Object Name
End		

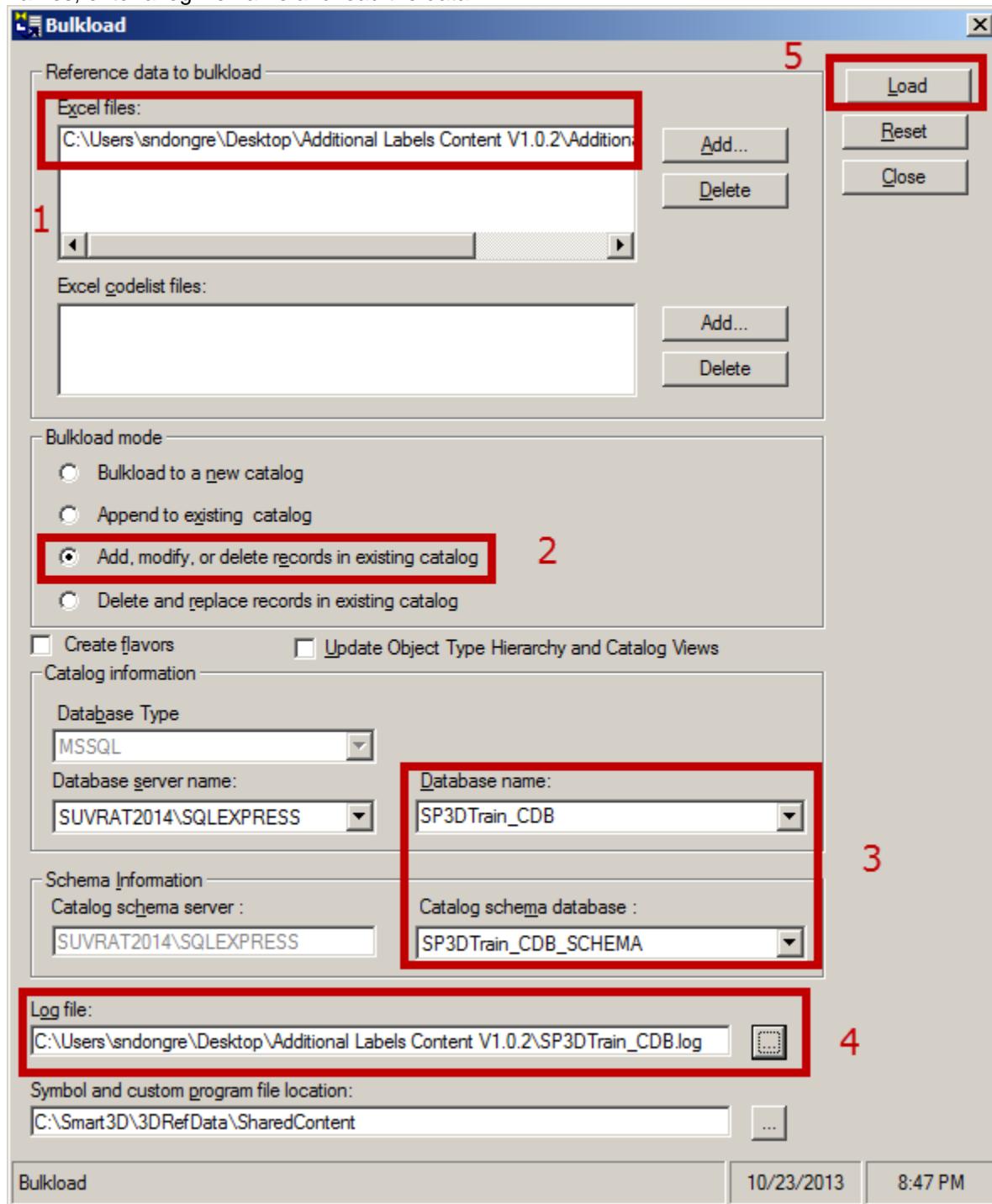
(Ctrl) ▶ Report ClassNodeType R-Hierarchy

4. Save the excel file
5. Follow the instructions in the PDF document to register the query dll

6. Start the Bulkload utility from Start → Programs → Intergraph Smart 3D → Database Tools → Bulkload Reference Data



- On the bulkload form, browse to the file, pick the bulkload mode, select the database and schema names, enter a log file name and load the data.

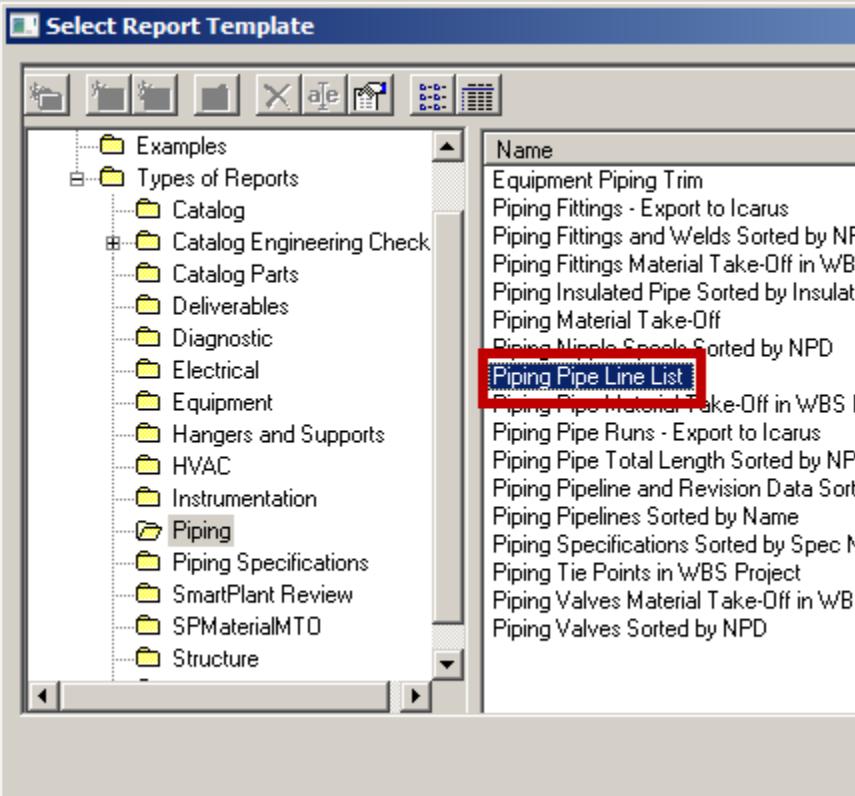


Embedding Labels

Pipeline Length Report using Run Length Label

Create Report

1. Switch to Drawings and Reports task
2. Right mouse click on 'Spreadsheet Reports' and select 'Create Report'
3. Select the 'Piping PipeLine List' report and click OK.



4. Select Finish on the Report Parameters dialog.
5. Edit Template on the report
6. Add a CONSTANT of type String with value [Run Length]

CONSTANT	Length	String	[Run Length]
----------	--------	--------	--------------

7. Enter Design Layout
8. Type the word Length in cell N1 and use format painter to apply same format as one of the header cells
9. Drag the Length property to cell N4

N	O
Length	
#PipeLine!Length#	

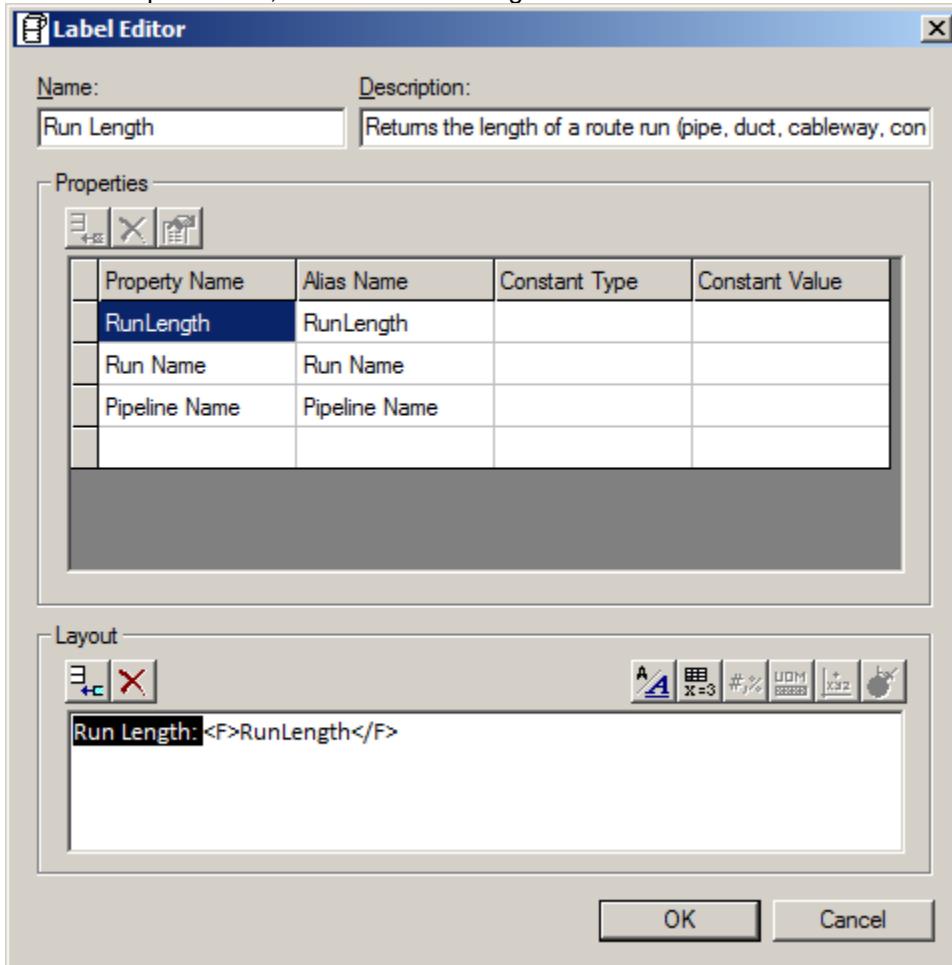
10. Close excel and save when prompted
11. File → Save Report Template
12. Update the report.

13. Open the updated report. Note that the run length is a string property reported as direct output of the label.

Paint Spec	Op. Temp	Op. Pressure	Length
Metallic coating			Run Length: 1328 mm
Metallic coating			Run Length: 0 mm
Metallic coating			Run Length: 273 mm
Metallic coating			Run Length: 2810 mm

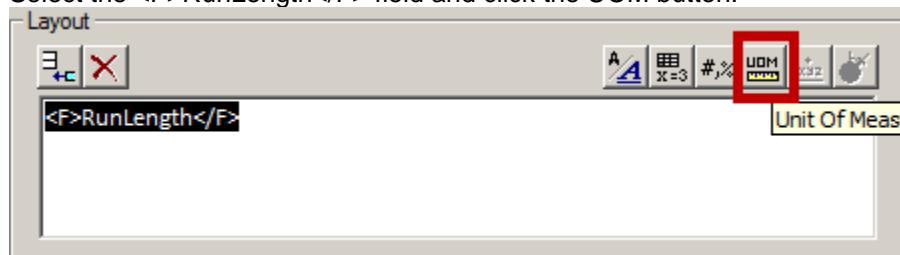
Edit Label

1. Switch to Catalog task
2. Tools → Define Label
3. In the Examples folder, locate the Run Length label and edit label.

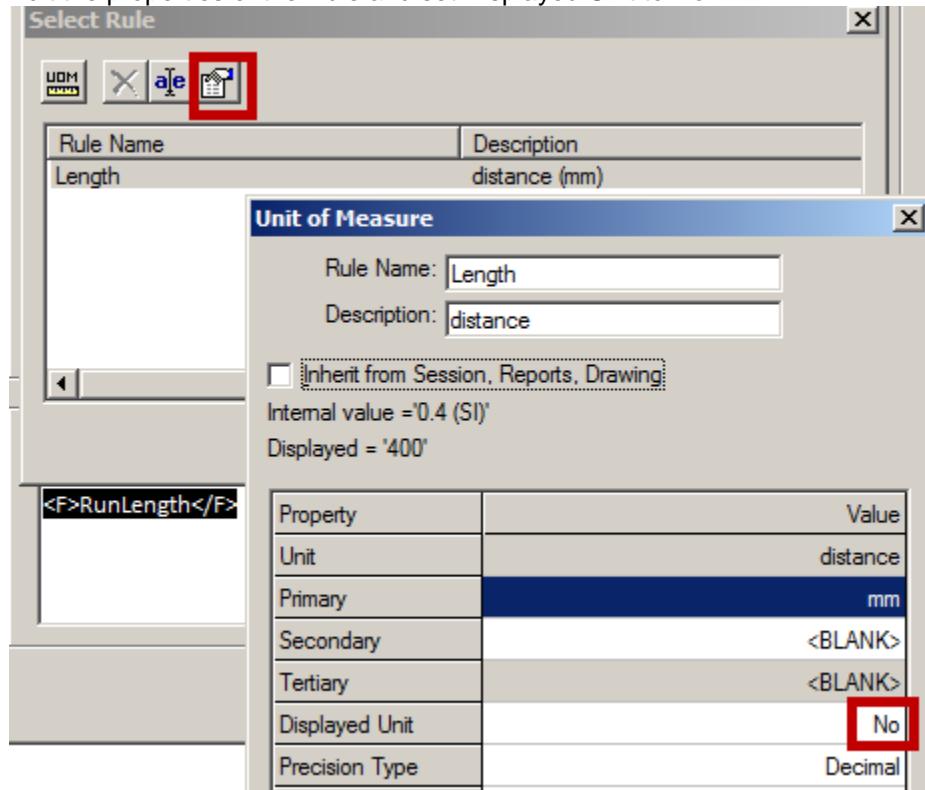


4. Delete the text 'Run Length: ' from the Layout

5. Select the <F>RunLength</F> field and click the UOM button.



6. Edit the properties of the Rule and set Displayed Unit to No



7. Click OK to save the rule
8. Click OK to select rule
9. Click OK to save the label
10. Exit the session

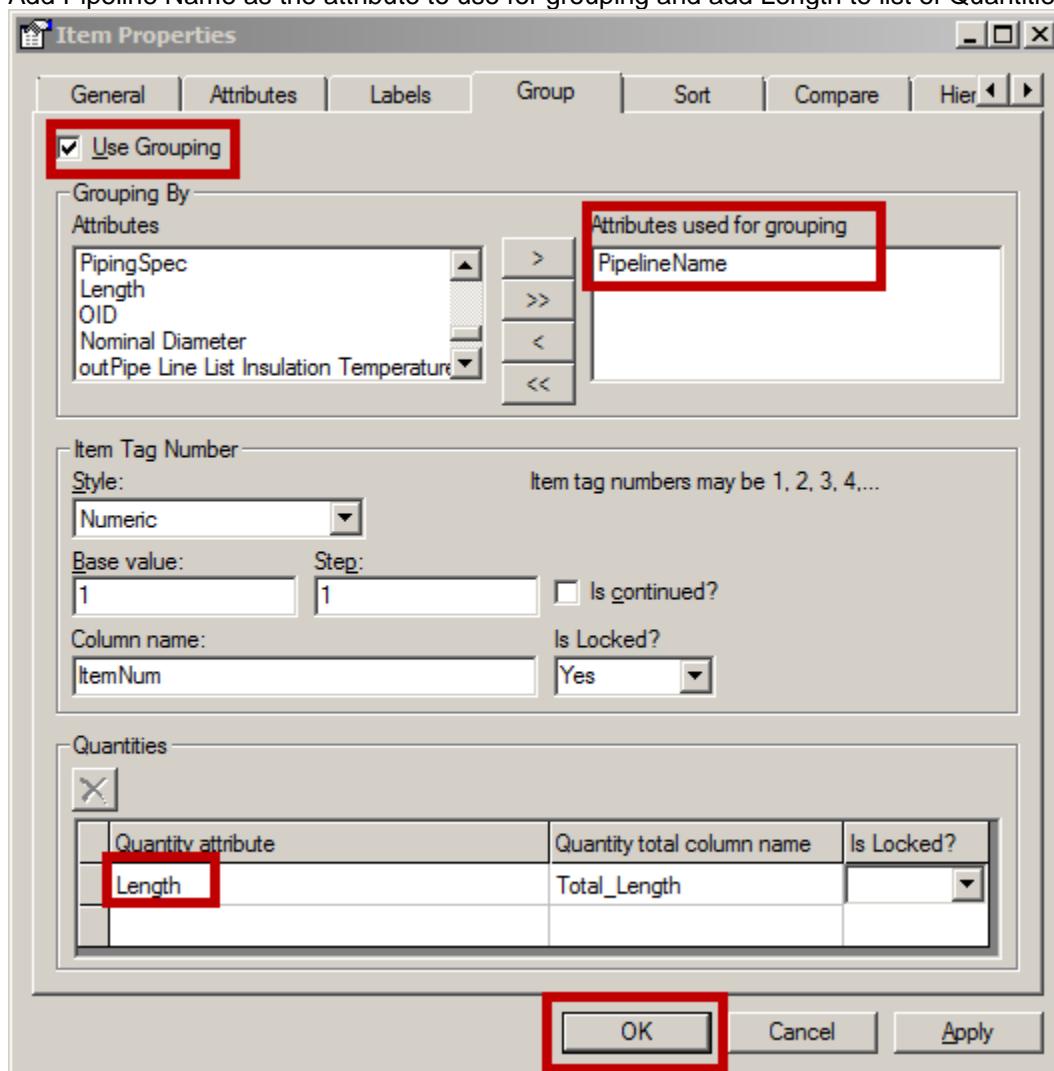
Edit the report

1. Start Smart 3D, switch to the Drawings and Reports task and re-update the report.

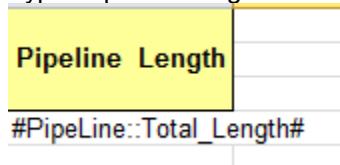
2. Open the report, note that now the length is simply a number

K	L	M	N
Paint Spec	Op. Temp	Op. Pressure	Length
Metallic coating			1328
Metallic coating			0
Metallic coating			273
Metallic coating			2810
Metallic coating			0
Metallic coating			1404
Metallic coating			

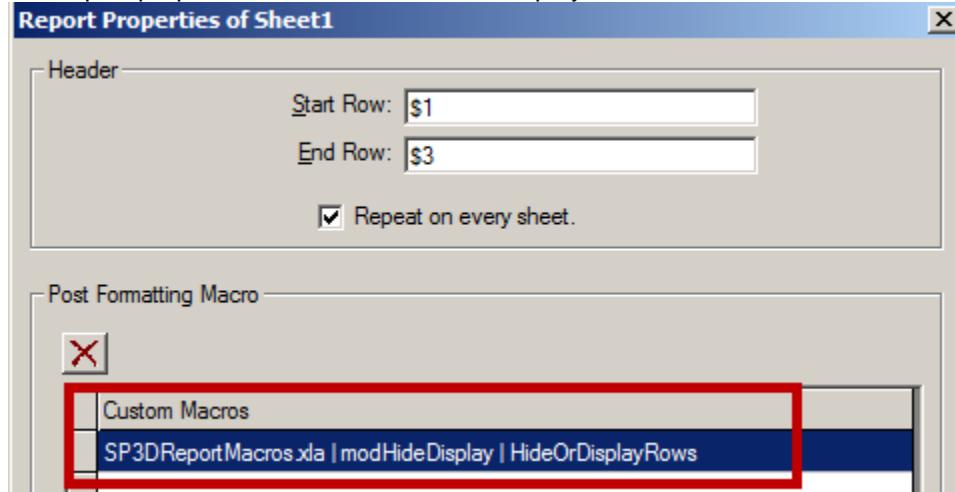
3. Edit template, switch to design layout and edit properties on the Pipeline Query.
 4. Switch to the grouping tab and check the box to use grouping
 5. Add Pipeline Name as the attribute to use for grouping and add Length to list of Quantities.



6. Type 'Pipeline Length' in cell O1, drag the Total_Length attribute to cell O4.



7. Edit report properties and add the HideOrDisplayRows Macro



8. Save excel and save report template

9. Update the report

10. Open the report. Notice that total length for each pipeline is reported in column O.

Pipeline	Pipeline Length
1001-P	10845
1002-P	19995
2001-P	18606
2002-P	33767
2003-P	35154
2004-P	2374
2005-P	18288
300-W	41862
301-W	11564
303-W	12619
311-W	42493
400-P	32994
401-P	17946
402-P	22739
403-P	8927

Embedding output of a label in SQL query

1. Edit template on the report 'Equipment Report SQL Query'

2. Edit the query as below to include the name of embedded label in []

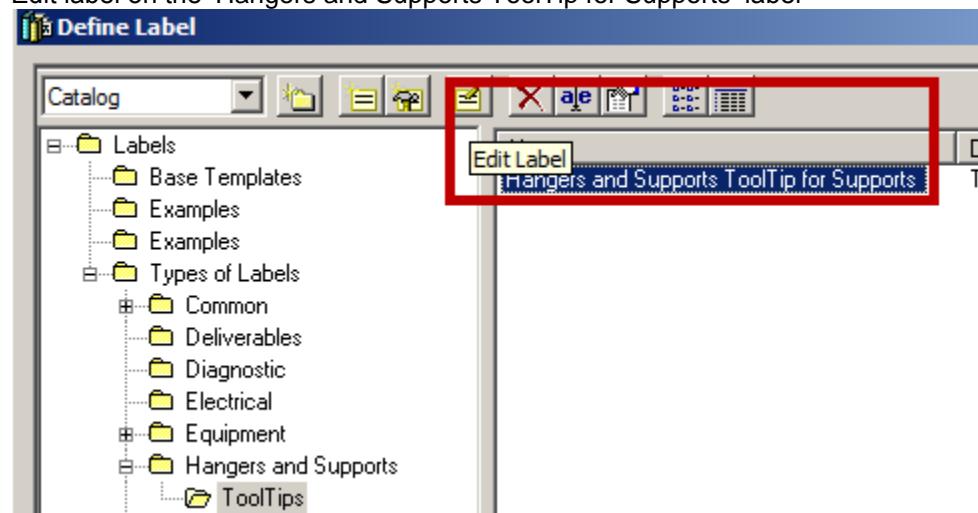
```
SQLQuery2.sql ...ndongre (57)* SQLQuery1.sql - not connected* Objec
select j2.Oid as 'OID',
j1.ItemName as 'Parent Name',
j2.ItemName as 'Equipment Name',
'[Equipment Bearing]' as Bearing
from AnasEqpAschid x1
join JNamedItem j1 on j1.Oid = x1.OidOrigin
join JNamedItem j2 on j2.Oid = x1.OidDestination
```

3. Use the green triangle to run the query and verify syntax
4. Enter design layout and drag the Bearing value to cell G9, type 'Bearing' in the header row
5. Close excel, save when prompted and save report template
6. Update the report and observe the bearing is reported for each equipment

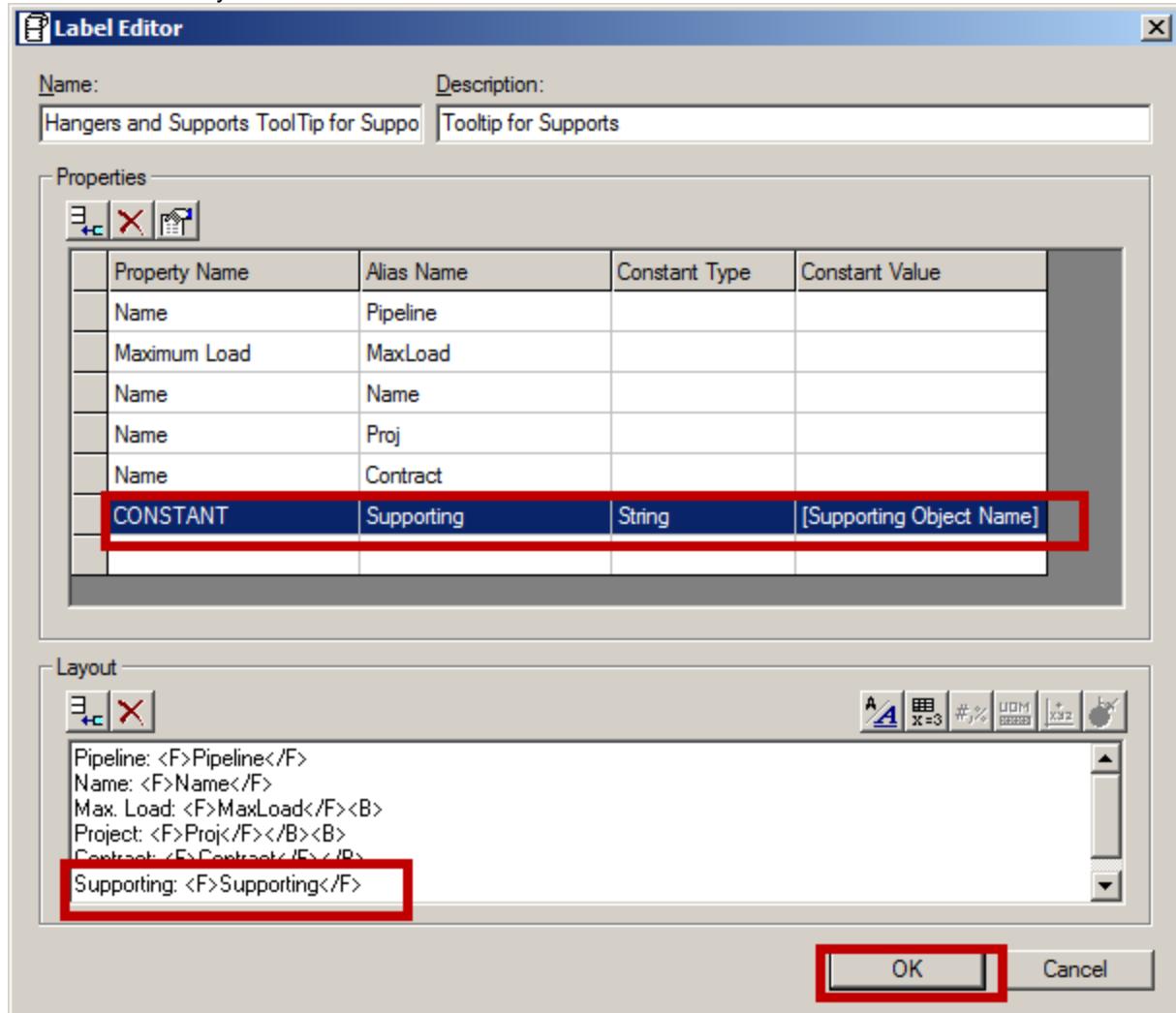
Parent Name	Equipment Name	Bearing
Equipment	T-101	90. deg
Equipment	P-101	90. deg
Equipment	B-101	90. deg
CT	LG-1	180. deg
CT	LG-5	0 deg
CT	LG-6	0 deg
CT	LG-7	0 deg
CT	LG-8	0 deg
CT	LG-4	180. deg
CT	LG-2	180. deg
CT	LG-3	180. deg

Embedding one label into another

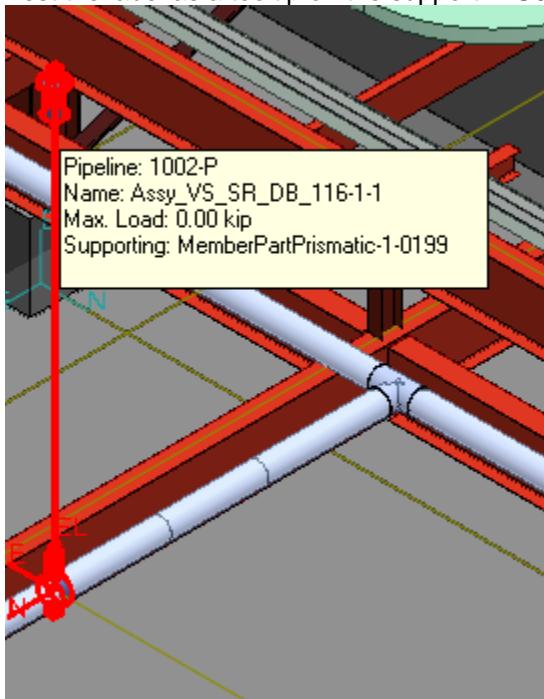
1. Switch to Catalog task
2. Tools → Define Label
3. Edit label on the 'Hangers and Supports ToolTip for Supports' label



4. Add a row in the Properties tab to add the '[Supporting Object Name]' as a CONSTANT of type String with Alias Name 'Supporting'
5. Add the field to Layout



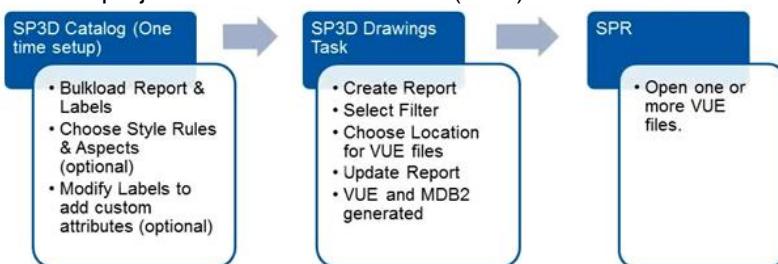
6. Test the label as a tooltip on the support in U01



S3DToSPRDirect Reports

Introduction

SPR Direct enables a user to publish S3D data and objects (DGN/DWG, PDS, and Reference 3D) directly to graphics (VUE) and label data (MDB2) files. These files can then be opened in SmartPlant Review. SPR Direct provides a SmartPlant Review spreadsheet report type in S3D that contains labels used to generate the attribute information for SmartPlant Review. This method offers an easier, faster way to get the S3D project to SmartPlant Review (SPR) where it can be interactively reviewed and analyzed.



Features

In addition to faster data exchange, SPRDirect provides the following features:

- A WYSIWYG operation in that the attributes seen in S3D can likewise be seen in SmartPlant Review.
- The process of adding or removing attributes is simplified
- There is no longer the need for additional tools for the data exchange:
 - Schema Editor is not needed for mapping properties on the S3D side
 - The SP3D Label Selector and the UOM Converter utilities that come with SmartPlant Review are not required.

SPRDirect Components

SPRDirect provides the following components or tools used to complete the process:

- SmartPlant Review spreadsheet report (S3DtoSPRDirect) – Creates the graphic (VUE) file and the attribute data (MDB2) file. The attribute data that goes into the MDB2 file is controlled by the S3D labels. This report is delivered in the SharedContent folder.
- MapClassIDToLevelDisciplines.txt – Contains the S3D class ID-to-SmartPlant Review Discipline and Level mapping. It is used to obtain the SmartPlant Review discipline and level information for the graphic objects while the graphic (VUE) file is being created. This file contains the mapping relations between S3D classes and SmartPlant Review disciplines. Edit this file **ONLY** if there is a need to change a name or a level number for the generated VUE file.
- ReportSettings.txt – Provides the S3D class-to-label mapping. Edit this file if there is a need to add or remove custom labels, and define additional surface style rules and aspects.
- Report template and Seed file (Seed.mdb2) – Creates the MDB2 and VUE files.

Output Files

Files that are generated from the update process are listed below. The Project file name is the name of the filter that was associated with the report during parameter selection. These files are automatically created into a user-defined location also defined in the parameter selection.

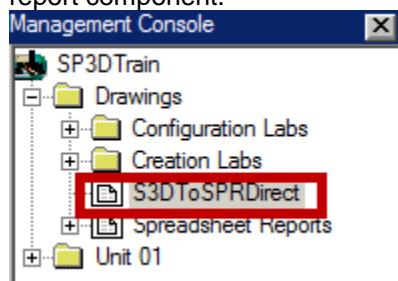
- *[Project File name].log*
- *[Project File name]TimeLogRelease.txt*
- *[Project File name].vue*
- *[Project File name].mdb2*

Creating S3DtoSPRDirect Report

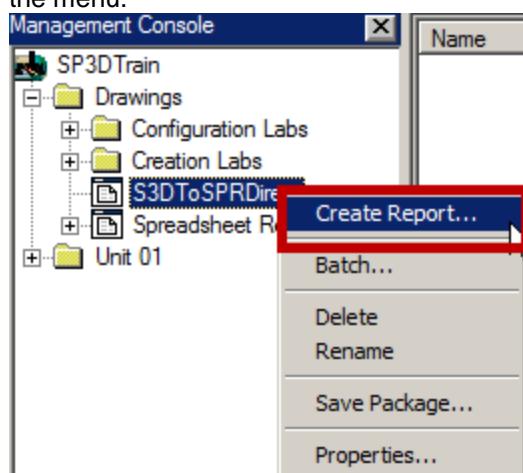
Objectives:

- Create a new report in the “Drawings and Reports” task of S3D
- Generate files from S3D that can be used in SmartPlant Review

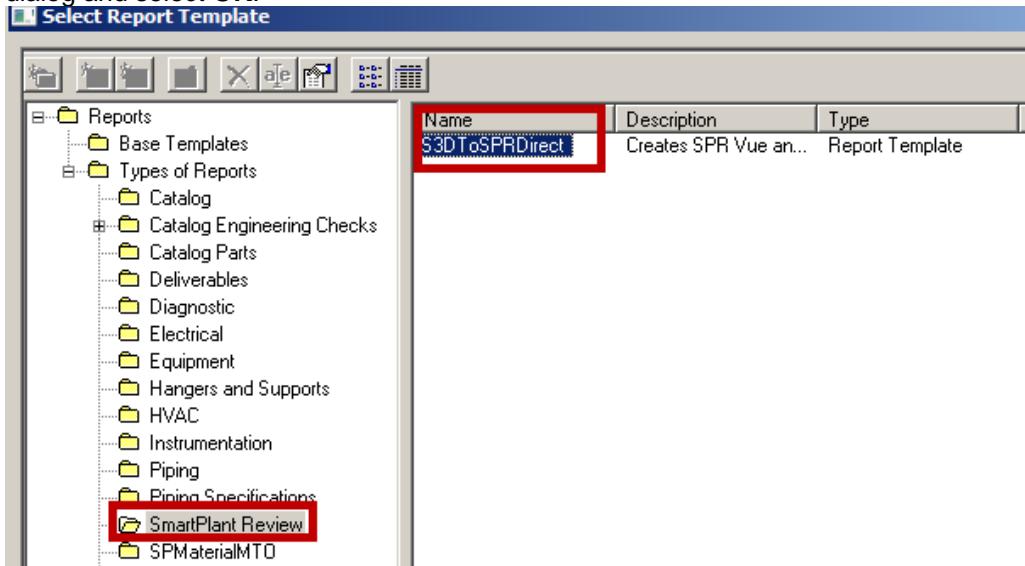
1. Switch to the “Drawings and Reports” task.
2. In the **Management Console**, expand the “Drawings” folder and locate the “S3DtoSPRDirect” report component.



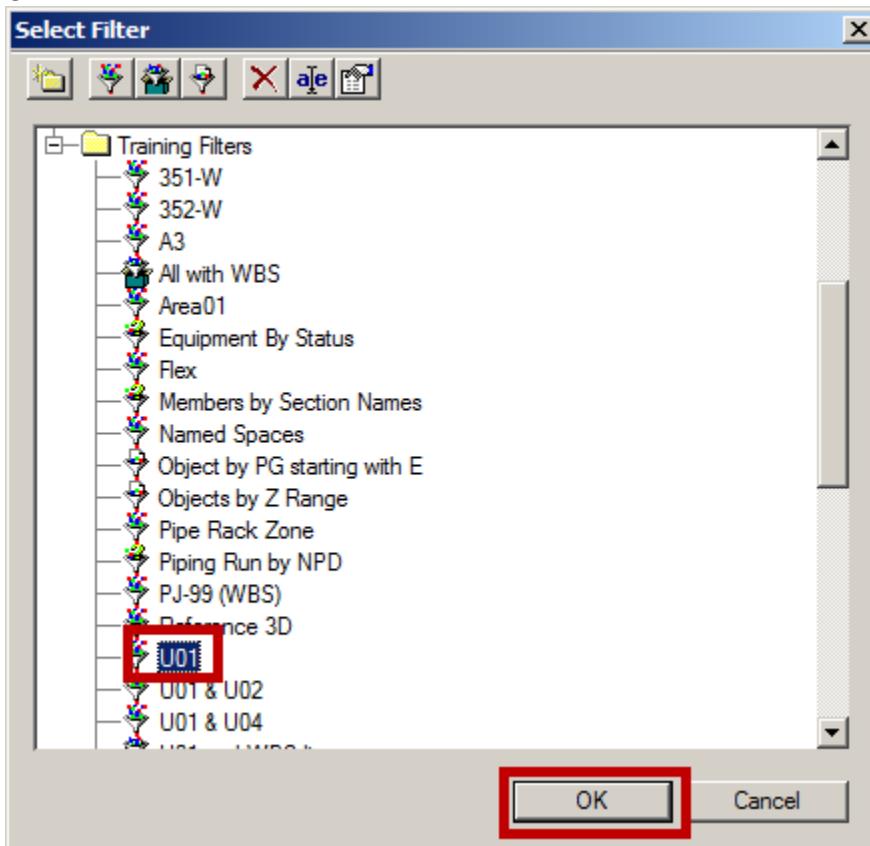
3. Right mouse click on the “S3DtoSPRDirect” report component and select “Create Report” from the menu.



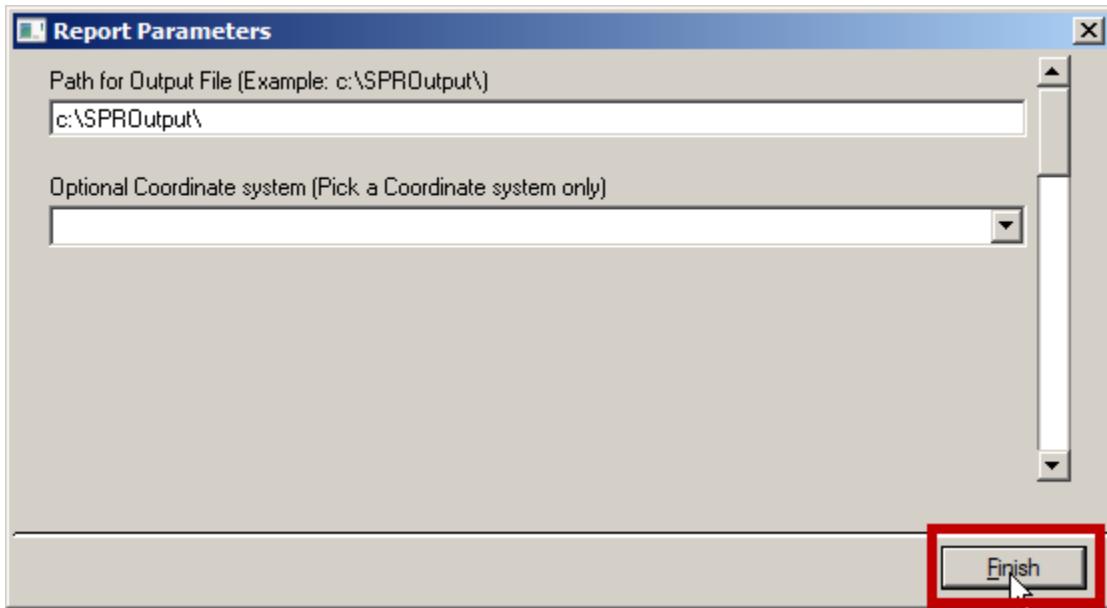
4. From the “Select Report Template” dialog, navigate to **Reports > Types of Reports > SmartPlant Review** and select the “S3DToSPRDirect” report template from the right pane of the dialog and select **OK**.



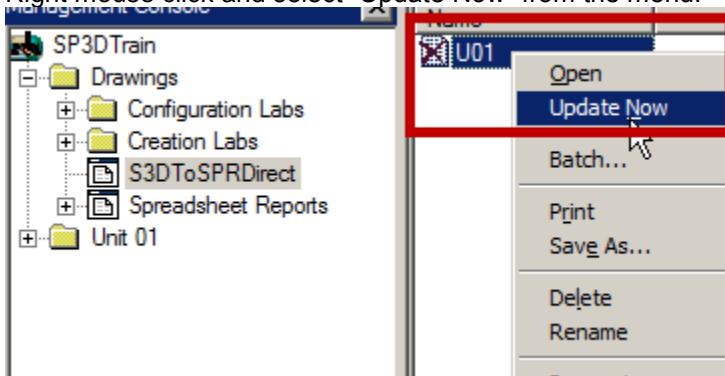
5. Next, the “Select Filter” dialog will open. This filter will determine the area that the .vue and mdb2 files will be generated from. In **Plant Filters > Training Filters** select the “U01” filter and select **OK**.



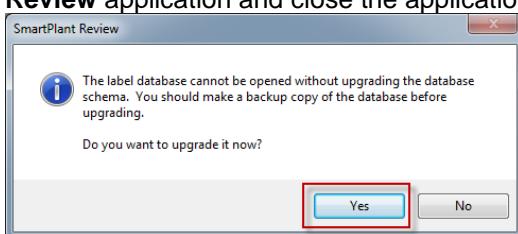
6. Select **Finish** on the Report Parameters dialog.



7. The report will be created in the document list. Right mouse click on the newly created report and select 'Rename'
8. Rename the report to 'U01'
9. Right mouse click and select "Update Now" from the menu.



10. As the report updates, open "Windows Explorer" and navigate to C:\SPROutput and notice the files that have been created. When the report has completed its update, the "U01.vue" file will be created in this folder location.
11. Once the "U01.vue" file is created, double click the file to open it in the **SmartPlant Review** or **SmartPlant FreeView** product.
12. As the application opens, accept any dialogs that appear. View the results in the **SmartPlant Review** application and close the application when finished.



Copy S3DToSPRDirect report to Catalog

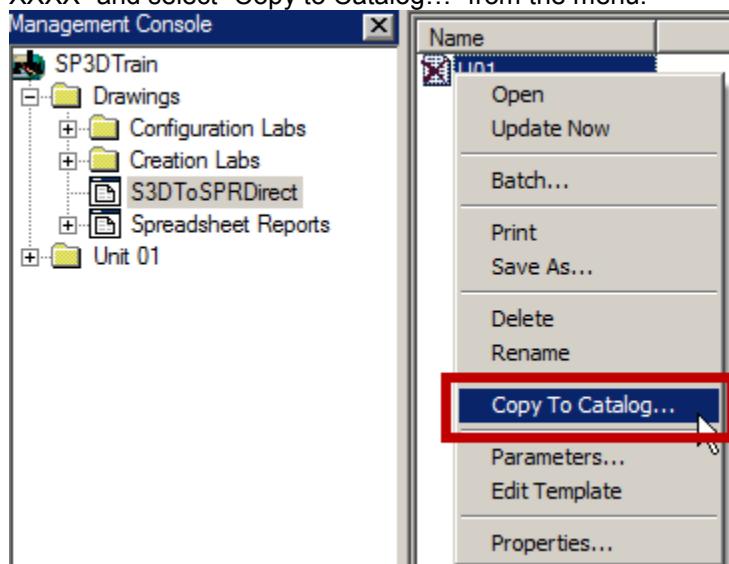
It is possible to have multiple copies of the report in the catalog, each with its own “ReportSettings.txt” file. This allows the user to have additional types of SPR vue file output. For example, report 1 would apply a set of style rules based on the pipeline fluid code for use as an internal review. Report 2 could apply a set of style rules based on a construction status and could be used for a client review.

Objective:

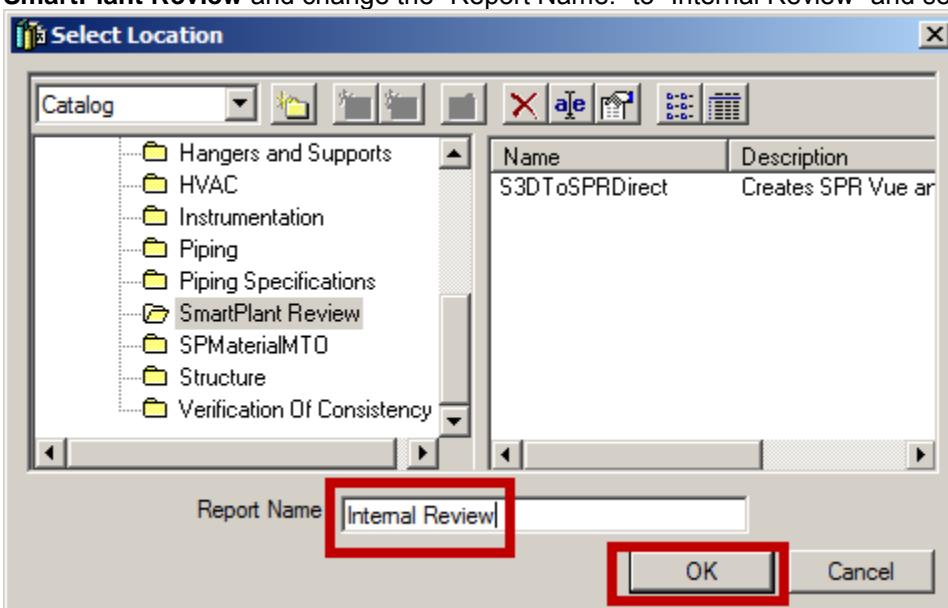
- Copy the delivered S3DToSPRDirect report to the catalog.
- Copy other, necessary files to the new reports location in SharedContent.
- Make necessary changes to xml files in order to use the desired ReportSettings.txt file.

Copy to Catalog

1. In the **S3D** application, change the task to the “Drawings and Reports” task, if not already there.
2. Right mouse click on the report (U01) created in the previous lab, and select “Edit Template” from the menu.
3. Once in the report’s template, select **File > Save Report Template** from the main menu bar.
4. If prompted, select the same filter on the “Select Filter” dialog from **Plant Filters > Training Filters > U01** and select **OK**. On the “Report Parameters” dialog, select **OK**.
5. From the **Management Console**, select the “S3DtoSPRDirect” report component created in the previous lab. This will close the report template.
6. On the report in the “Document List”, right mouse click on the report “S3DTOToSPRDirect-hsv-XXXX” and select “Copy to Catalog...” from the menu.

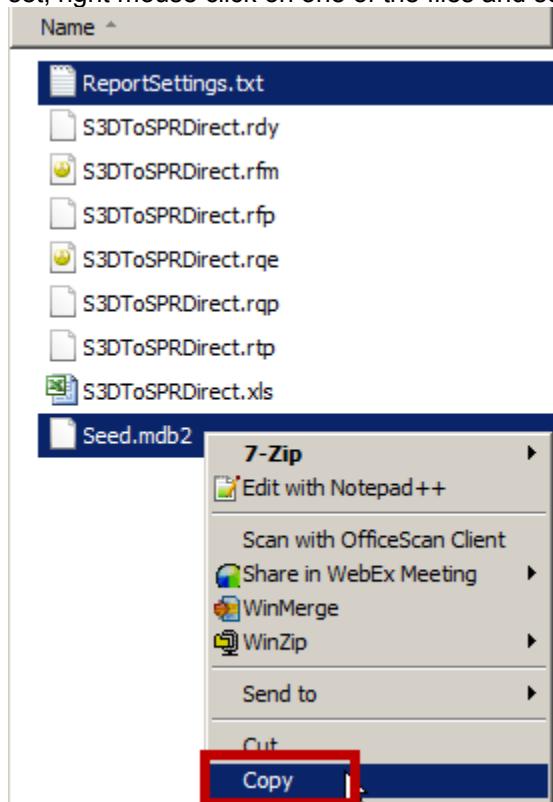


7. On the “Select Location” dialog, expand the folder list to **Reports > Types of Reports > SmartPlant Review** and change the “Report Name:” to “Internal Review” and select **OK**.

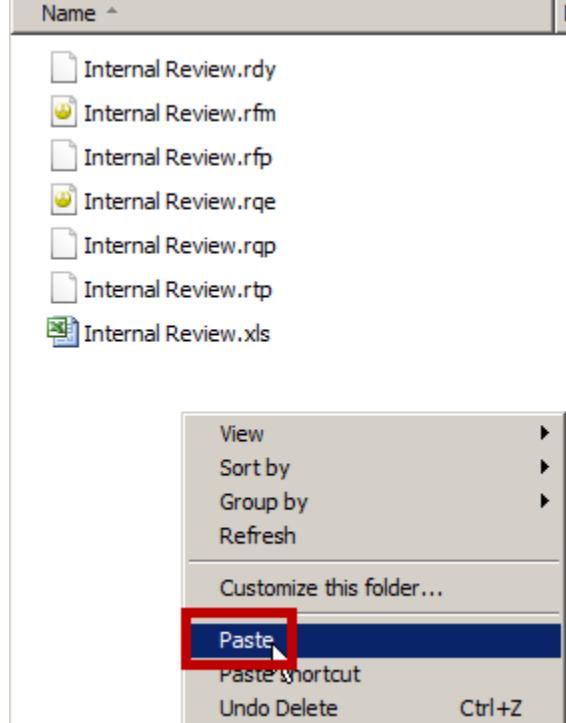


Copy Additional Files

1. Now that the report is in the catalog, some files need to be copied from the original report (S3DtoSPRDirect) it was created from. Minimize the **S3D** application and open **Windows Explorer**. Navigate to the **SharedContent\Reports\Types of Reports\SmartPlant Review** folder.
2. Open the folder “s3dtosprdirect” and select the “ReportSettings.txt” file. Hold down the **Ctrl** key on the keyboard and left mouse click on the “Seed.mdb2” file. With these two files in the select set, right mouse click on one of the files and select “Copy” from the menu.

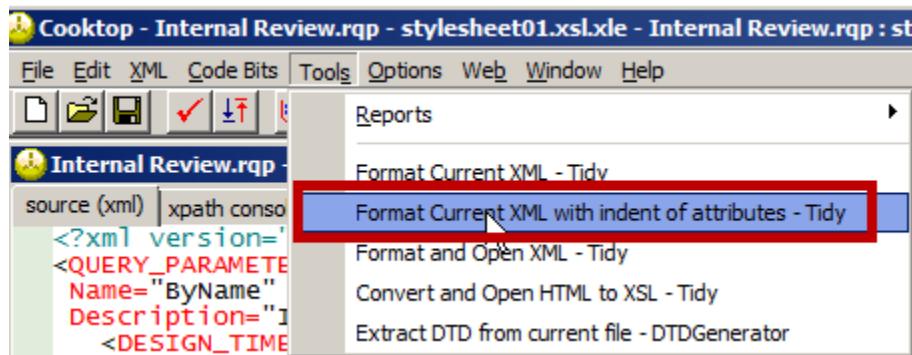


3. Once the files are in the copy buffer, navigate back one folder and double click on the “Internal Review” folder. Right mouse click in this window and select “Paste” from the menu.



Change File Path for the “ReportSettings.txt”

1. In the “Internal Review” folder, open the “Internal Review.rqp” file in the **Cooktop** or other XML editor.
2. If using **Cooktop**, select **Tools > Format Current XML with indent of attributes – Tidy** from the main menu.



3. Scroll to the bottom of this file to the last “FIELD” element and locate the “DefaultValue” field. Change this value to point to the new location of the “ReportSettings.txt” file. This path should be **Reports\Types of Reports\SmartPlant Review\Internal Review\ReportSettings.txt**. Refer to the image below:

```

<FIELD
  Name="ReportSettings"
  Index="3"
  DataType=""
  AllowMultiSelect="No"
  IsHidden="Yes"
  IsRequired="No"
  IsSelectableObject="No"
  Caption="ReportSettings.txt file location (Example: Reports\Types of Reports\smartPlant Review\S3DToSPRDirect\ReportSettings.txt)"
  LookUp="Text"
  SQLType="BStr"
  DefaultValue="Reports\Types of Reports\smartPlant Review\Internal Review ReportSettings.txt"
  DisplayedValue=" "/>

```

4. Save and close the document after this edit is completed.

Editing ReportSettings.txt

The ReportSettings.txt configuration file provides the S3D ‘class name-to-label name’ mapping and the surface style rule mapping. This file is installed in the ..\SharedContent\Reports\Types of Reports\SmartPlant Review\S3DToSPRDirect folder.

The ReportsSettings.txt file must be edited when:

- Additional aspects to objects are added in the S3D project file.
- New user defined labels and surface style rules need to be added.

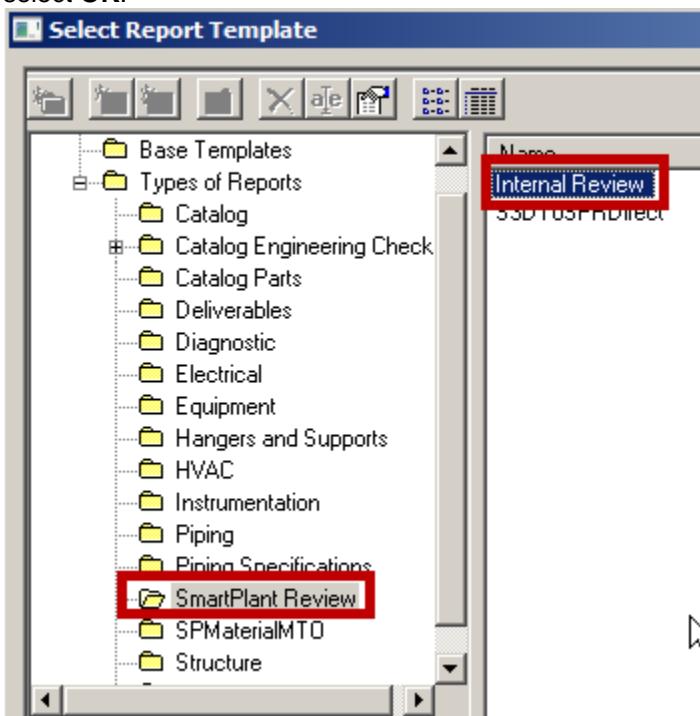
Adding Aspects

1. In the **Windows Explorer** window open the “ReportSettings.txt” from the “Internal Review” folder (the **Windows Explorer** window should already be opened to the “Internal Review” folder from the previous lab).
2. With the “ReportSettings.txt” file opened in **Notepad**, scroll down to the bottom of the file and locate the “Aspects” section. Left mouse click after the word “Insulation” and select the “Enter” key on the keyboard to create a new line. Type “Maintenance” on this new line.

```
! Surface style rules are applied in the order they appear in the list below - first one wins
[SurfaceStyleRules] ! Hardcoded row do not remove

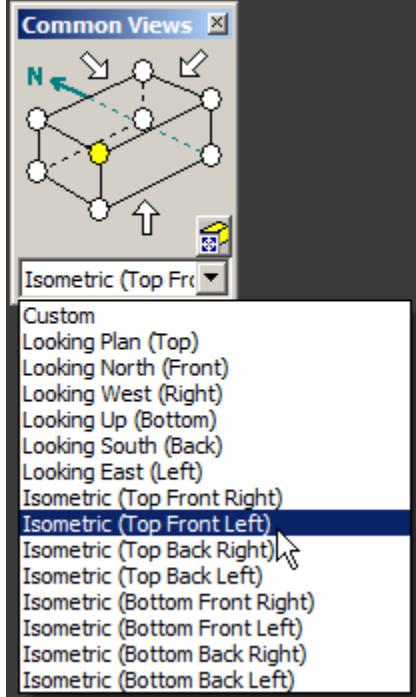
! Graphics belonging to all aspects listed below will be output to the vue file
[Aspects] ! Hardcoded row do not remove
Simple Physical
Insulation
Maintenance
```

3. Save and exit out of the “ReportSettings.txt”.
4. In the **S3D** application, right mouse click on the “S3DToSPRDIRECT” report component in the “Management Console” and select “Create Report...” from the menu.
5. In the “Select Report Template” dialog, select the newly created report “Internal Review” and select **OK**.

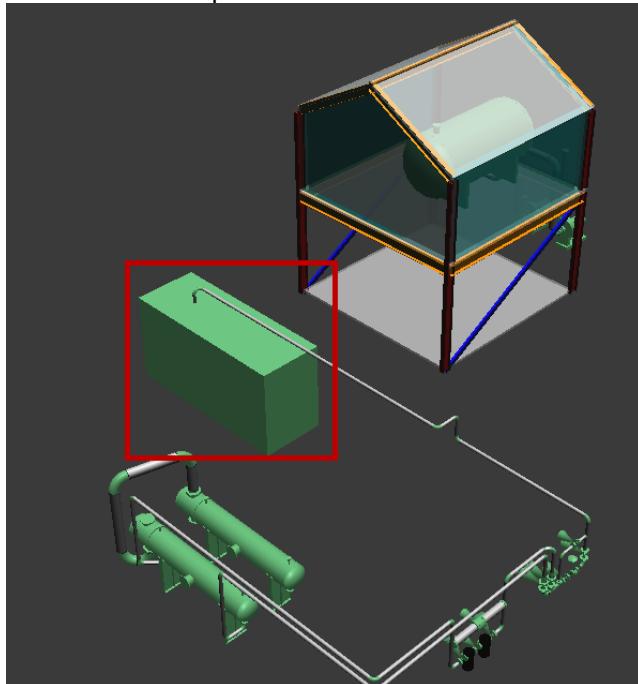


6. In the “Select Filter” dialog, select the “U04” filter under **Plant Filters > Training Filters** and select **OK**.
7. On the “Report Parameters” dialog, click **Finish**.
8. Rename the newly added report to ‘U04’
9. Right click on the new report “U04” and select “Update Now” from the menu.
10. Once the report has completed updating, open **Windows Explorer** and navigate to “c:\SPROoutput” folder and double click the “U04.vue” file to open in the **SmartPlant Review** application.

11. As the application opens, accept all confirmations on the dialogs. After the application has opened, select the “Common Views” command  from the main toolbar and drop the list down on the “Common Views” dialog and select “Isometric (Top Front Left)” from the list.



12. From the main toolbar, select the “Fit View to Model” command .
13. Notice the box that appears for the vessel in the middle of the view. This is showing the “Maintenance” aspect of the vessel in the **SmartPlant Review** session.



14. Exit the **SmartPlant Review** application.

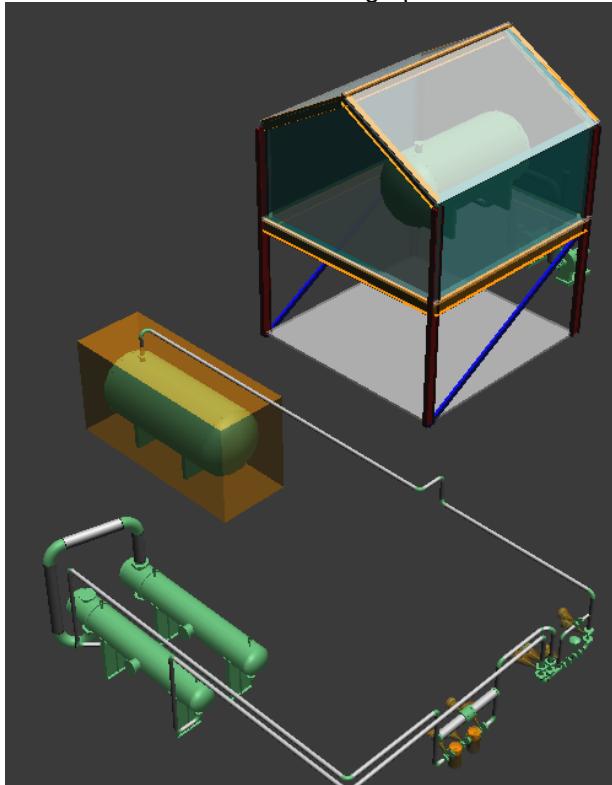
Adding Surface Style Rules

1. In the **Windows Explorer** window open the “ReportSettings.txt” from the “Internal Review” folder (the **Windows Explorer** window should already be opened to the “Internal Review” folder from the previous labs).
2. With the “ReportSettings.txt” file, opened in **Notepad**, scroll to the bottom of the file and locate the end of the row “[SurfaceStyleRules] ! Hardcoded row do not remove” and select the “Enter” key on the keyboard to create a new line. Type “00Maintenance Aspects”. This entry should look as follows:

```
! Surface style rules are applied in the order they appear in the list below - first one wins
[SurfaceStyleRules] ! Hardcoded row do not remove
00Maintenance Aspects
```

! Graphics belonging to all aspects listed below will be output to the vue file
[Aspects] ! Hardcoded row do not remove

3. Save and exit out of the “ReportSettings.txt”.
4. In the **S3D** application, right mouse click on the report “U04” and select “Update Now” from the menu.
5. Once the report update is complete, open **Windows Explorer** and navigate to “c:\SPROutput” folder and double click the “U04.vue” file to open in the **SmartPlant Review** application.
6. As the application opens, accept all confirmations on the dialogs.
7. Notice the Maintenance aspect added in the previous lab has changed to “Translucent Yellow”. Some of the valves also show graphics in maintenance aspect.



8. Exit the SmartPlant Review application.