

Case Details - 11067285

Short Description	Customer reports that updating formats on drawing Templates changes all parameters to have ":D" and no longer reads part/asm parameters.
Symptoms	Drawing template mode does not function as expected if the parameters referenced in the format already exist in the drawing template.
Steps to Reproduce	<p>1.) Create a new format with tables referencing both designated and Windchill system parameters.</p> <p>2.) Create a new, empty drawing template. Check this into Windchill, and then check it back out. It now will contain all of the required designated and Windchill system parameters.</p> <p>3.) Put the drawing into "template" mode.</p> <p>4.) Apply the previously created format to the drawing.</p> <p>All of the parameters that are supposed to be mapped to the future model instead snap to the already existing drawing parameters.</p> <p>How can a template be built that displays both drawing parameters and model parameters with the same name? Re-applying the format after a model is added to the drawing will fix the problem, but that then defeats the whole point of making a template.</p> <p>I did attempt to temporarily add a model to the template, replace the format, and then delete the model. (This will be necessary to create the drawing program.) Unfortunately, when deleting the model, all the referenced parameters in the tables (that pointed to the model) are deleted as well.</p> <p>Applications - Template mode needs to work for all parameters, not just ones that don't exist in the drawing. 95% of all parameters we reference on a drawing are the model's values, not the drawing's values. The template needs to reflect this.</p> <p>Here are the other documents I have already referenced:</p> <p>https://www.ptc.com/appserver/cs/view/solution.jsp?n=111848</p> <p>https://www.ptc.com/appserver/cs/view/solution.jsp?n=33558</p> <p>https://www.ptc.com/appserver/cs/view/solution.jsp?n=CS51521</p> <p>https://www.ptc.com/appserver/wcms/standards/ssl/frefull_cskdb.jsp?im_dbkey=87775</p>
Reason for Closure	Currently not supported
Closure Details	As per the article CS125598 and SPR 1938925 and TAN 147551 customer's requirement goes as an enhancement. Requested customer to submit a Product Idea @PTC Community

Related Documents

Type	Number	Description
Article	CS125598	When Creating a Drawing Template, if a Parameter that Exists in the Template is Called out into a Table Cell, the Table Cell Will Append ":D" to the Parameter Callout Even if the User Edits out the ":D" out.
SPR	1938925	When creating a drawing template, if a parameter that exists in the template is called out into a table cell, the table cell will append ":D" to the parameter callout even if the user edits the ":D" out. This makes it impossible to call out certain Windchill and Pro/INTRALINK related parameters onto a Drawing Template.

Comment created by Tom on 10-Oct-2012 10:36:29 EDT

For the record, here is a summary from our web meeting.

The fundamental problem is that even in template mode, when formats and tables are added to a drawing if the parameter being referenced in the format or table matches the name of an already existing parameter in the drawing, the format and table will automatically snap to the existing drawing parameter instead of waiting for the future model's parameter to be read.

For example, if the drawing has a parameter called "REVISION" and the future model has a parameter called "REVISION", every time a table with &revision is added to the drawing its syntax will automatically be changed to &revision:d. This happens regardless of whether or not the drawing is set to "Template" mode. This makes it impossible to have a drawing with a pre-existing, designated parameter named "REVISION" and at the same time place a table on the drawing that will read the "REVISION" parameter from the future model. One would think that by being in template mode and using the syntax <parameter>:d for the drawing and <parameter> for the model would allow the one without the :d to stay unmapped until a model is added, but unfortunately both are immediately changed to :d.

Comment created by Seshu (PTC) on 16-Oct-2012 19:51:46 EDT

Informed the customer that when a parameter ¶m_name is called in a drawing format Table or note, irrespective of the current mode (ie., Drawing or Template), Creo would first search the active drawing model for the parameter value and then in the drawing. If the parameter exists in the drawing, the parameter value will be populated by referring the drawing parameter value thus replacing the text ¶m_name with ¶m_name:D.

Advised the customer to call model parameters manually using the drawing model session number

Comment created by Tom on 29-Oct-2012 11:31:38 EDT

I have read and re-read TPI 105936 and TPI 114055. With much testing there still appears to be several problems.

First, TPI 105936 will not work correctly if multiple models are in session. Using the :o syntax won't work if the drawing's model doesn't have a session ID of o (which is very likely with multiple models open). It's not realistic to tell users that they must remove all models from session every time they want to create a drawing program.

Second, TPI 114055 says to use the drawing program syntax "param_name:d=param_name" and then states the following: "Program cannot be executed.' is returned. This is because the parameter on the right side of the equation cannot be found in the active model. If the parameter could be found, then Pro/ENGINEER will automatically place the drawing session id after this parameter". This is not true if the parameter already exists in the local drawing. This brings me to the third problem.

Third, the drawing program is not "searching" for valid parameters the way the previous solution for Case 11067285 stated. The solution said, "Informed the customer that when a parameter ¶m_name is called in a drawing format Table or note, irrespective of the current mode (ie., Drawing or Template), Creo would first search the active drawing model for the parameter value and then in the drawing. If the parameter exists in the drawing, the parameter value will be populated by referring the drawing parameter value thus replacing the text ¶m_name with ¶m_name:D." This previous statement is correct if the parameter does not exist in the model but does exist in the drawing. However if the parameter DOES exist in the drawing the table or note will point to the model in preference to the drawing. This however is NOT the case with the drawing program. If the same parameter exists in BOTH the drawing and the model (as directed by TPI 105936), and the drawing program syntax "param_name:d=param_name" is used (as directed by TPI 114055), the drawing program will NOT connect the

drawing parameter to the model parameter but will instead change the syntax to “param_name:d=param_name:d”. This should not happen if the model is searched for a valid parameter FIRST before the drawing (as stated in TPI 114055).

Email Communication on 17-Apr-2013 14:58:34 EDT

From: [Seshu](#)

After going through your requested scenario several times again and again, Following are the options I could see to meet your requirement.

Requirement: Auto Create drawing parameters with values driven from the associated solid model and have the drawing parameters designated to be populated in PDM.

Possible Solutions:

1. Use session ID 0 or 1 in the Drawing program. This will allow to create parameters in the drawing template and designate them. Can be used in WC. Need to erase all the files from memory and have only a single part or assembly in session before creating the drawing.
2. Do not use session ID - Use a mapkey that will create a drawing and designate the required params after the drawing creation. Drawing creation, Designate params can be a part of one Mapkey and pause mapkey for the template selection. The Drawing template should be located on local disk not in Wind Chill PDM.
3. Use Model check to designate params before saving, so that users will be forced to designate set of params- Need to check whether the list of std parameters can be designated in one click and use Model check gateway to prevent drawings from checked in without having certain set of parameters designated.

Please go through these possibilities and let me know your comfortable time. I'll call you to discuss this issue.

Email Communication on 17-Apr-2013 17:06:51 EDT

From: [Tom](#)

Thanks for your work on this issue. Let me attempt to clarify the requirements.

Drawing Template Requirements:

- A. Must contain all Windchill parameters and company specific parameters
- B. Must have all parameters designated so Windchill can see them
- C. Must contain a drawing program to link the values of the future model parameters to the drawing parameters
- D. Must contain tables on the template that display the values of BOTH drawing and model parameters with the SAME name (ex. PTC_WM_VERSION)
- E. Must be stored in Windchill (No Pro/e data is managed outside of Windchill, including formats/templates/etc.)

Now, let me show you the problem that occurs.

In order to pre-create tables that display the value of drawing parameters in the template (requirement D), these parameters must already exist in the drawing. That's fine, since we want them to be designated anyway, so let's

create and designate all drawing parameters (requirements A and B). Now let's create all tables (in "template" mode), save, and store in Windchill. At this point we've met requirements A, B, D, and E, but NOT C. If we now attempt to take this drawing template and add a drawing program to it, the program will immediately snap all program lines from "PARAMETER:D = PARAMTER:o" syntax to a "PARAMTER:D = PARAMETER:D" syntax (which is worthless). Even though the drawing does not have a model (and is in template mode), since the parameter being referenced by the ":o" syntax also exists in the drawing in the ":D" form, the drawing program immediately substitutes the ":o" for ":D". This is the core issue.

All the other ideas are simply workarounds to deal with this core problem with the drawing program. Unfortunately each of these proposed solutions create bigger problems. Here are the issues with each method (my comments after each suggestion).

1. Use session ID 0 or 1 in the Drawing program. This will allow to create parameters in the drawing template and designate them. Can be used in WC. Need to erase all the files from memory and have only a single part or assembly in session before creating the drawing.

Problem: Pre-building a template with both a drawing program and all designated parameters and then storing it in Windchill won't work. See the core issue above.

2. Do not use session ID - Use a mapkey that will create a drawing and designate the required params after the drawing creation. Drawing creation, Designate params can be a part of one Mapkey and pause mapkey for the template selection. The Drawing template should be located on local disk not in Wind Chill PDM.

Main Problem: Drawing tables cannot be pre-built on the template to refer to both model parameters and drawing parameters (the whole point of a template).

Secondary Problems: Template can't be stored in Windchill (or parameters will already exist). The drawing program has to be manually created every time the template is used. The drawing parameters have to be manually designated (after their creation by the drawing program) every time the template is used.

3. Use Model check to designate params before saving, so that users will be forced to designate set of params- Need to check whether the list of std parameters can be designated in one click and use Model check gateway to prevent drawings from checked in without having certain set of parameters designated.

The issue isn't just designating the parameters. The drawing parameters need to exist in the template so they can be used in the tables, so they can be pre-designated, and so the drawing program has something to connect to. Unfortunately, if the parameters do exist in the drawing, and the corresponding model parameters do not, they syntax gets all messed up when the program runs.

Here is my summary statement from an earlier comment:

If the same parameter exists in BOTH the drawing and the model (as directed by TPI 105936), and the drawing program syntax "param_name:d=param_name" is used (as directed by TPI 114055), the drawing program will NOT connect the drawing parameter to the model parameter but will instead change the syntax to "param_name:d=param_name:d". This should not happen if the model is searched for a valid parameter FIRST before the drawing (as stated in TPI 114055).

Recap

· Creating the drawing program after drawing creation from a template is not a good solution.

a) The whole point of a template is to have all this stuff already in there (notes, tables, views, parameters, program, etc.)

b) The drawing program can only be created with one model in session at a time. (Not even remotely practical. No one is going to close their assemblies and all erase all models every time they want to make a new drawing.)

· Creating the drawing parameters after drawing creation is not a good solution either.

a) The template can't contain tables that refer to these (missing) parameters.

b) The template cannot be stored in Windchill.

c) The parameters cannot be pre-designated.

In my mind, the required fix is to make drawing programs follow "template mode". They must not execute their program until AFTER a model has been added to the drawing (just like tables).

Sorry this is so long. Thanks for continuing to work on it.

Comment created by Tom on 17-Apr-2013 17:39:47 EDT

Actually, I should change that last line. The original reason for creating this case is that drawing tables don't obey template mode when a format is re-applied to the drawing. The issue with drawing tables and drawing programs is the same. Both get their syntax messed up if the drawing has no model, which of course is exactly what a template is! Here is what I should have said:

The required fix is to make both drawing programs and drawing tables obey "template mode". They must not search for or replace any model parameter (one that does not have a :D after it) with a drawing parameter (even if one exists) until AFTER a model has been added to the drawing and searched first.

Email Communication on 17-Apr-2013 19:01:21 EDT

From: [Seshu](#)

Thanks for your patience and detailed explanation of your view on the reported issue. As per your old and latest comments, the functionality you are requesting has been already reported to R&D through SPR 1938925 <<https://www.ptc.com/appserver/cs/view/spr.jsp?n=1938925>> and was concluded that this functionality is working as designed and to be able to change this behaviour, an enhancement should be planned/submitted. I would request you to go through TAN 147551 <<https://www.ptc.com/appserver/cs/view/solution.jsp?n=147551>>.

With the above information, I am trying to find a most nearest possible solution for your scenario based on the practice that is being followed by other customers as well rather than this is enhanced once you post this requirement in Ideas@Planet PTC and reviewed by our Product Management Team.

I am still trying to understand why you are saying this is not possible? "Problem: Pre-building a template with both a drawing program and all designated parameters and then storing it in Windchill won't work. See the core issue above." Whereas, I am seeing it working as expected. The only limitation is that I need to have only one part or one assembly in session during new drawing creation.

One key point I would like to highlight is that the change of a drawing program line from "PARAMETER:D = PARAMTER:o" syntax to a "PARAMTER:D = PARAMETER:D" happens only when the drawing parameter already exists in the drawing by the time drawing program lines are written.

Please find the attached example template drawing and part. If you create a new drawing using this template for the

given part, the drawing parameter test will have its value controlled by the part parameter test value and is pre-designated. Let me know if I am missing something.

Another option is Option 2, which is suggested regarding use of mapkey is again one of the standard practice that is being followed by customers from Intralink days.

Once again thanks for your patience on this issue.

Email Communication on 18-Apr-2013 16:26:28 EDT

From: Tom

Thank you for the prompt and detailed response. My response is below. (It ended up being the length of a small novel.) Please don't take anything personal, I really do appreciate your patience on this.

1. SPR 1938925<<https://www.ptc.com/appserver/cs/view/spr.jsp?n=1938925>> is "not customer viewable". Can you please do what is necessary to allow me to view this SPR?

2. TAN 147551<<https://www.ptc.com/appserver/cs/view/solution.jsp?n=147551>>

a. From the resolution section, "Templates were never intended to have the capability of avoiding calling out a parameter that exists in the drawing so they can get it from the solid when/if the solid shows up." Per WF5 documentation, "Drawing templates contain three basic types of information for creating new drawings. ... The third type is parametric notes. Parametric notes are notes that update to new drawing model parameters and dimension values. The notes are re-parsed or updated when the template is instantiated." Sure seems like that's exactly what we are talking about here. Creating parametric notes specific to the model that update when a model is added (and not getting these mixed up with the ones in the drawing).

b. The first workaround, "make the note before putting the drawing in the PDM system" will not work. I demonstrated this to you when we had the WebEx session. Even if the tables/notes are created before the drawing parameters exist, any attempt to use the template while connected to Windchill will mess the syntax up. As soon as the template is loaded into session (behind the scenes) the Windchill system parameters become available and immediately the syntax gets changed before the model is added to the new drawing. This will only work if the template is created outside of Windchill, stored outside of Windchill, and used outside of Windchill. We do not store anything outside of Windchill and we're certainly not going to connect and disconnect from Windchill every time we want to create a drawing.

c. The second workaround, "have a parameter ... in the template for the solid" will only work if you pre-build a second set of ALL parameters (that you may ever want to use in a drawing) in ADVANCE of trying to create the drawing. This is nuts. We have over 2 million files in our vault. There is no way we are going to go back and re-add these extra parameters to all of these models. Creation with a mapkey isn't practical either due to how relations get created from mapkeys (completely overwriting any existing relations).

d. <Rant On>

I'm not blaming you, but TAN 147551<<https://www.ptc.com/appserver/cs/view/solution.jsp?n=147551>> feels like PTC is just making excuses for failing to properly integrate Pro/e with their PDM environment. Of course templates were designed before the PDM system. But if PTC is going to promote the connectedness and interoperability between Windchill and Pro/e, then they need to fix these things, not just say, "it was never intended to work this way". That can be said about Pro/e in its entirety. Pro/e was never originally designed to work as part of a PDM system, but it does now through software additions and changes. What is "intended" needs to be updated. I just don't get why PTC would choose to only half integrate one product with the other. When they are made aware of an issue with using two of their own products together, they should fix it.

</Rant Off>

3. "I am still trying to understand why you are saying this is not possible? ... Whereas, I am seeing it working as expected. The only limitation is that I need to have only one part or one assembly in session during new drawing creation." The issue is not just creating a new drawing from the template, it's that the template can't be created in the first place. You cannot create a template with a table on it having both model parameters and drawing parameters (with the same name), have a drawing program linking the model parameters and drawing parameters (with the same name), and have the drawing parameters designated. You can't build a fully functional template with everything in it.

4. "One key point I would like to highlight is that the change of a drawing program line from "PARAMETER:D = PARAMETER:O" syntax to a "PARAMETER:D = PARAMETER:D" happens only when the drawing parameter already exists in the drawing by the time drawing program lines are written." Bingo. That's the whole problem. In a Windchill environment the drawing program gets run (behind the scenes) BEFORE a new model gets added. Even if you have an empty template with no parameters (just a program) as soon as you attempt to use it while connected to Windchill, the drawing program will create the drawing parameters BEFORE the model is added and the syntax will get messed up.

5. "Please find the attached example template drawing and part. If you create a new drawing using this template for the given part, the drawing parameter test will have its value controlled by the part parameter test value and is pre-designated. Let me know if I am missing something." It works because the drawing program and table are referring to a parameter that does not exist in the drawing, and won't ever, even after the drawing has been saved. This is not the case with company designated parameters and Windchill system parameters that exist in EVERY cad document as soon as it's loaded into session.

Fundamentally, TAN 147551<<https://www.ptc.com/appserver/cs/view/solution.jsp?n=147551>> correctly summarizes the problem, but it provides no real solution. Identical parameters DO exist in both the drawing and the model. The whole point of TPI 114055<<https://www.ptc.com/appserver/cs/view/solution.jsp?n=114055>> is to synchronize parameters of the SAME name between a drawing and a model in a template. These recommendations will not work in a Windchill environment because it's not possible to create and use a template without the parameters automatically being created. As soon as the drawing is loaded into session it's going to have the parameters, like it or not. They WILL exist before any drawing program is ever executed or a model is added.

My goal in all this is to get PTC to fix templates so they obey

CS51521<<https://www.ptc.com/appserver/cs/view/solution.jsp?n=CS51521>> for parameters that exist in both the model and the drawing, for both tables and drawing programs. The only workaround I have is to both reapply the format and create the drawing program after the drawing has been created, and after the model has been added. The big limitation here is that only one model can be in session while these steps are being performed. I shouldn't have to erase everything from memory every time I need to create a new drawing. The whole point of the template is to have this stuff already in there.

Email Communication on 01-May-2013 16:21:42 EDT

From: [Seshu](#)

Thanks for the latest comments in the case. I did understand your requirement. However, was trying to put across the best possible options that are currently being used. I did noticed that the SPR 1938925<<https://www.ptc.com/appserver/cs/view/spr.jsp?n=1938925>> I've referred previously was not Customer viewable. So, I've made this Viewable now.

As per the article CS125598<<https://www.ptc.com/appserver/cs/view/solution.jsp?n=CS125598>> and SPR 1938925<<https://www.ptc.com/appserver/cs/view/spr.jsp?n=1938925>> and TAN 147551<<https://www.ptc.com/appserver/cs/view/solution.jsp?n=147551>> your requirement goes as an enhancement. So, please submit a Product Idea @PTC Community<https://www.ptc.com/appserver/cs/misc/PTC_Product_Ideas.jsp>