



Bentley Plant Workshop

2013 Bentley LEARNing Conference

PLOP2WK1

OpenPlant 2D/3D Interoperability

Team Leader: Tony DeRosa

Team Members: Brian Earles

Bentley Systems, Incorporated
685 Stockton Drive
Exton, PA 19341
www.bentley.com

This page left intentionally blank.



Table of Contents

| | |
|---|-----------|
| Preface | 5 |
| Session Description..... | 6 |
| Learning Objectives | 6 |
| Chapter 1: 2D to 3D Interoperability..... | 7 |
| Overview | 7 |
| 2D to 3D Interoperability..... | 7 |
| Basic Process..... | 7 |
| Details of the Consistency Checker Tree | 7 |
| Defining Properties Required for Consistency..... | 8 |
| Limitations of 2D to 3D Interoperability..... | 8 |
| Chapter Summary..... | 9 |
| Chapter 2: 2D to 3D put into Practice..... | 11 |
| Overview | 11 |
| Creating a model file..... | 11 |
| 📄 Exercise 1: Launching a Working Model and Referencing in a PID i-model..... | 11 |
| Inserting Components | 11 |
| 📄 Exercise 2: Inserting Components from the Consistency Checker Tree..... | 11 |
| 📄 Exercise 3: Inserting a Component from the Place Component Dialog | 12 |
| 📄 Exercise 4: Inserting Pipelines using the Consistency Checker. | 13 |
| Chapter Summary..... | 13 |
| Chapter3: Defining a Property for Consistency | 15 |
| Overview | 15 |
| Examining the Schema..... | 15 |
| 📄 Exercise 1: Loading the schema in the class editor | 15 |
| Working with Consistent Properties..... | 15 |
| 📄 Exercise 1: Messing with the Consistency Checker | 15 |
| Chapter Summary..... | 16 |
| Learning Paths | 17 |
| Assessment | 18 |

This page left intentionally blank.



Preface

In this workshop, you will get a first pass hands on look at the new 2D to 3D interoperability tools in OpenPlant Modeler. Understand that this is the first release of this evolving technology.

You will learn how to reference in an i-model of the PID and you will use the resulting Consistency Checker node in the Item Browser to:

Insert components from the Consistency Checker.

Insert components from the Browse Tag tool within the component insertion dialog.

Learn how to create an instance of a pipeline from the Consistency Checker node.

Look at the OpenPlant_Supplemental_ConsistencyChecker schema and see how to set a property so that it must be the same in 3D and 2D in order to be considered consistent.

Please remember to take the course materials with you as well as the i-model of the PID used in this manual.

Note Prerequisite Knowledge Level: Participant should have a basic understanding of plant design principles and have a basic understanding of OpenPlant Modeler and the Item Browser to Check Out components and place equipment.

SESSION DESCRIPTION

Participants in this session are taught how to work with process flows that support P&IDs and 3D models using OpenPlant.

LEARNING OBJECTIVES

After this course you will be able to:

- Create 3D OpenPlant elements from a referenced PID
- Learn how to make a property a requirement for consistency
- Understand the limitations of the current 2D to 3D Interoperability

Note At the end of this training session, an assessment will be given. We will review all assessment questions and answers to see what you have learned.

NOTE YOU CANNOT DELETE THIS PAGE! We need it for our accreditation, so please fill out as indicated!



Chapter 1: 2D to 3D Interoperability

OVERVIEW

In this chapter you will learn the basics of creating 3D elements from 2D PID i-models. You will start with an overview of the basic process. You will also learn the current limitations of the process.

2D TO 3D INTEROPERABILITY

OpenPlant 2D to 3D Interoperability lets you compare components in an OpenPlant PowerPID drawing with the equivalent components in an OpenPlant Modeler drawing. It allows you to place components from a referenced PowerPID i-model from the Item Browser tree. This functionality lets you exchange tag information as well as perform consistency between function and physical design.

The Primary functions are to assist in automating the process of creating OpenPlant 3D models from OpenPlant PowerPID drawings and assist in maintaining that consistency between the two systems.

BASIC PROCESS

The basic process begins by publishing an i-model from the desired OpenPlant PID drawing. In OpenPlant Modeler a working files created.

At this point you use the default MicroStation reference command and reference in the PID i-model.

When the i-model is referenced in a new node is added to the Item Browser. This new node is the Consistency Checker. Multiple PID i-models can be referenced into the same working model and a new Consistency Checker node will be added for each PID.

DETAILS OF THE CONSISTENCY CHECKER TREE

When a PowerPID i-model is attached the Consistency checker node is added to the standard Item Browser and will display the components in from the PowerPID i-model. As stated before, multiple i-models can be attached simultaneously displaying multiple Consistency Checker trees.

In each tree each component will be color coded and supplied with a status indicator.

The status indicators are:

- **Green** with a check mark  – the component has been placed in the model.
- **Red** with minus sign  – indicates a component has not been placed in the model.
- **Blue** with a warning sign  – indicates a component has been placed in the model but has inconsistencies between the 2D and 3D component.



DEFINING PROPERTIES REQUIRED FOR CONSISTENCY

Another feature of the 2D to 3D Interoperability is the ability to define specific properties that must match in order for the 2D and 3D component to be considered to be consistent with each other.

When the i-model is attached to the 3D model, OpenPlant looks for certain specified 2D properties to compare to the matching 3D property. If these properties match then the model is found to be consistent with the PID. Adding additional properties to this consistency criteria can be performed by the user by modifying the `OpenPlant_Supplemental_ConsistencyChecker` schema. You will review this later in this workshop.

LIMITATIONS OF 2D TO 3D INTEROPERABILITY

This is the first release of an evolving technology. As such it does have some limitations. These include the following:

- 2D nozzle placement in OpenPlant Modeler is not supported.
- Control Valve tag name convention is by default different in both applications. You will need to update the EC Calculated Expression or be aware of the change in the Device Type Code property.

- YOU MUST use OpenPlant Modeler SELECT series 5 and OpenPlant PowerPID SELECT series 5 Refresh build 202.

CHAPTER SUMMARY

In this chapter, we have learned the basics of the 2D to 2D consistency checking available in OpenPlant. You have learned that you must reference in an OpenPlant PowerPID i-model in order to verify consistency. You learned how to read the Consistency Checker tree to determine the status of components. You learned that specific properties can be added to the schemas to define what makes a 2D component consistent with the 3D component. Lastly you learned of the current limitations of the process.

This page intentionally left blank.

Chapter 2: 2D to 3D put into Practice

OVERVIEW

In this chapter you will put into practice what you learned from Chapter 1. You reference in a PID i-model and insert components in OpenPlant Modeler from the Consistency Checker tree and using the Browse for tag option in the respective component dialogs. Lastly you will look the results when a property that is required to be the same in the 2D and 3D environments is different, the results in the Consistency Checker tree and how to correct the inconsistency.

CREATING A MODEL FILE

In this section you will load a model and reference in a PID i-model.



Exercise 1: Launching a Working Model and Referencing in a PID i-model

Exercise Objective: In this exercise you load an OpenPlant Working model and reference in a PID i-model.



1. From inside **ProjectWise Explorer** and the **Models>Working Models** folder launch the **WM1** working model file.
2. After OpenPlant Modeler loads, open the **Item Browser** if not already open.
3. Pick the **References** tool.
4. Pick the **Attach Reference** command and browse to the **Drawings>OpenPlant P&ID's** folder.
5. Select the **PIDP01.dgn.i.dgn** file. Pick **OK**.
6. In the **Reference Attachment Settings for...** dialog, deselect the **Display** button under the **Toggles** section. Pick **OK**.
7. Close the **References** dialog.
8. Locate the **Item Browser** and expand the **Consistency Checker – PIDP01.dgn.i.dgn** node.
9. Expand the **Equipment** node. Note that the three equipment components are shown in red with the minus sign. This indicates that the components are not placed in the 3D model. In the next exercise you will start the process of making the 3D model match and be consistent with the 2D PID drawing.

INSERTING COMPONENTS

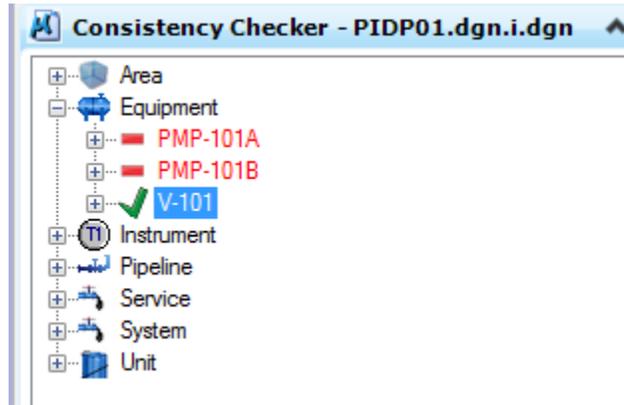
In this section you insert components from the Consistency Browser and Place Equipment dialog.



Exercise 2: Inserting Components from the Consistency Checker Tree

Exercise Objective: In this exercise you will insert equipment into the 3D model from the Consistency Checker tree

1. With the equipment node expanded right click on the **V-101** vessel and from the context menu select the **Place Par. Horizontal Vessel**
2. When the **Place Equipment** dialog loads note that the values from the PID have been transferred to the component in OpenPlant Modeler.
3. Place the vessel somewhere in the model.
4. Note that the tree refreshes and now **V-101** is shown with a green check mark indicating



that the 2D and 3D components are consistent.

Consistency Checker Tree showing V-101 consistent between 2D and 3D



Exercise 3: Inserting a Component from the Place Component Dialog

Exercise Objective: In this exercise you insert a piece of equipment from the Place Equipment dialog by using the Browse for Existing tag option.

1. From the **Equipment Task** select the **Place Horizontal Pump** tool.
2. In the **Place Equipment** dialog note the added **Tag Browser** category.



Tag Browser Category in the Place Equipment dialog

3. Pick the **Browse** button.
4. In the **Find Component** dialog expand the node **PIDP01,dgn.i.dgn**. Expand the **Vertical Centrifugal Pump** node. Pick **PMP-101A** tag in the list. Pick **OK**.
5. Pick a point in the model to place the pump.
6. In the Item Browser, at the top, pick on the **Refresh active tree** button.
7. Note that the **PMP-101A** pump is now shown with a green check mark indicating that the pump is consistent between the 2D PID and the 3D Model.



Exercise 4: Inserting Pipelines using the Consistency Checker.

Exercise Objective: In this exercise you learn how bring across the Pipelines from the PID to the 3D Model.

1. Expand the **Consistency Checker>Pipeline** node.
2. Right click on one of the pipelines.
3. From the context menu select **Create Pipeline**.
4. Note that the Pipeline is added to the model and can be located in the **Standard Preferences** and the **Pipeline Manager**.

Warning

The Create Pipeline command uses the PIPELINE and not the Pipe Run. The nominal size for the pipeline is not brought across because each run within the line could be of a different size. Note in the message center – “No nominal diameter was defined for pipeline created, defaulting to 0.5. This should be updated in Pipeline Manager.”

5. You should now set the pipeline to desired size and route your pipe.
6. You can now place inline components such as valves from the **Consistency Checker** as well as the **Place Component** dialog using the **Browse for Existing Tag** option.

CHAPTER SUMMARY

In this chapter you learned how access the PID tags by referencing in the PID i-model. You then learned how create equipment from the Consistency Checker tree and the Place Component dialogs. You also learned to create pipelines using the Consistency Checker tree.

This page intentionally left blank.

Chapter3: Defining a Property for Consistency

OVERVIEW

In this chapter you will look at the OpenPlant_Supplemental_ConsistencyChecker schema and learn how to make a property value a requirement for consistency.

EXAMINING THE SCHEMA

In this section you will load the OpenPlant_Supplemental_ConsistencyChecker schema in the Bentley Class Editor and review the component properties.



Exercise 1: Loading the schema in the class editor

Exercise Objective: In this exercise you will load the Class Editor and look at the schema property for equipment that is required to be consistent.

10. Pick **Start>All Programs>Bentley>Bentley Class Editor V8i.08.26>Bentley Class Editor V8i.08.26**
11. When the **Class Editor** loads open the **OpenPlant_Supplemental_ConsistencyChecker** schema located in the Project Workspace ...**Projects\OPModeler_Imperial\Dataset\Schemas** folder.
12. Expand the **Classes** node and select the **Equipment** class.
13. In the right hand panel pick the **Properties** tab.
14. Note the **OPERATING_PRESSURE** property is listed and that it has a **Consistency Checker Custom Attribute**. This **Custom Attribute** has a field "**Is Shared Property**" and this is set to **True**.

Note Adding the desired property and the adding the Custom Attribute to the property and setting its value to true allows this specific property on this component class that the value in the PID and the value for this property in the 3D model MUST MATCH in order for the components to be considered consistent.

15. **Exit** the Class Editor.

WORKING WITH CONSISTENT PROPERTIES

In this section you return to the model and make changes to the OPERATING PRESSURE property and see how the Consistency Checker reacts.



Exercise 1: Messing with the Consistency Checker

Exercise Objective: Make changes to a property so as to trigger the Consistency Checker into alerting the user that the 2D PID and 3D Model are not in sync.

1. If not already loaded, load the **Working** model from **ProjectWise**. Load the **Item Browser** and **reference** in the PID i-model.
2. Pick on the **Horizontal Vessel** and then right click and pick **Modify Component**.
3. Expand the **Operating Conditions** category and locate the **Operating Pressure** property.
4. Note that the property is 150 psig. This value was brought over from the PID when the equipment was placed in the model.
-  5. Change this value to **200**. Pick the **Apply Changes** button at the top of the dialog.
6. In the **Item Browser** pick the **Refresh active tree** button.
7. Note now that the **V-101** vessel is shown as being inconsistent with the PID.
8. Edit the vessel again and set the **Operating Pressure** back to **150** and apply the change.
9. **Refresh** the Item Browser tree and note that the vessel is now shown as being consistent between the PID and the 3D model.

CHAPTER SUMMARY

In this chapter you learned how to make a specific properties value be a requirement for consistency. You made changes to the value to see how the Consistency Checker displays a component when required properties do not match.



Learning Paths

All of our training through Bentley is accessible via learning paths, ready-made training plans that provide the recommended progression of skills training for a product, solution, job role, or experience level.

In developing your team, you can choose from Bentley-recommended learning paths or configure your own to meet your team's specific needs.

Learning paths help us:

- Develop our internal talent pool
- Make our employees more productive
- Prioritize learning topics by what is important to us as a company
- Reinforce our corporate culture of continuous learning

A sample Custom Learning Path can be found at the link below. This was created specific for this workshop.

<http://learn.bentley.com/app/Public/ViewLearningPathDetails?lpId=104096>



Assessment

1. ***The Consistency Checker can be run by referencing in a PID created in the “DGN” format?***

FALSE – The only format that can be used is an i-model.

1. ***Pipelines can be crated from Place Component Dialog?***

False - Pipelines can only be created using the Consistency Checker tree.

2. ***A component property cannot be used to force consistency between 2D and 3D components?***

True – A property can be used for this provided you add Consistency Checker Custom Attribute to the property and set its “Is Shared Property” field to true.

This page left intentionally blank.