Feature Specification – AutoCAD Grizzly

Validation

Last saved: 08/09/2007 7:17 AM

Author: Matt Denko

Table of Contents

Administration

1.1 Document History

Date	Name	Status	Comments
08/09/07	Matt Denko	Final	Posted final version. The following primary changes were made: • Added some additional detail to the markup updating behavior described in section 4.1.1.1.2 • Completed the description of various validation error conditions in section 4.1.1.1.3 • Added indication in Project Manager as to

Date	Name	Status	Comments
			whether a given validation error has previously been viewed or not – see section 4.2.3.3 • Added a new task dialog, "New Validation Errors detected", in section 4.2.4.3 • Removed various comment markers and highlighted sections from document, along with miscellaneous edits and deletions Change bars are enabled in the document so that new and modified content can readily be identified.
08/09/07	Matt Denko	Draft	Posted version that is 95% complete. Need to finish up the Error Summary section, and remove comments/highlighting in the document prior to final posting.
07/31/07	Matt Denko	Draft	Added section describing in greater detail validation markup behavior and the elements that comprise a markup. Also added initial mockups of new task dialog boxes. Reworked P&ID Validation Settings dialog box to include a number of additional options.
07/24/07	Matt Denko	Draft	Updated design to accommodate KKS workflows at Tuan-Ahn's request. Added an additional check for tag uniqueness. Also added back/fleshed out balloon notification mechanism and made some other changes/clarifications based on conversations with Jorge.
05/16/07	Matt Denko	Draft	At Jeff's request, turned attention back to finishing an initial draft of this document. Even though the feature is currently not in scope based on marketing requirements, the team thinks it would be useful to have SWD Delphi the feature. As a fairly complete implementation was already in place during the Guinness release but was scoped late in the development cycle, it might be worthwhile to Delphi the legacy design as well as the new one. Using the legacy design would require dropping some of the new requirements, or retrofitting the legacy design to accommodate them.
05/09/07	Matt Denko	Draft	Per Derry's requirements ranking from today, Validation is no longer a priority requirement for the release. At Jeff's request, tabled working on this document given current marketing priorities.

Date	Name	Status	Comments
05/07/07	Matt Denko	Draft	Shifting focus to 3D application.
04/10/07	Matt Denko	Draft	Starting with scoped Guinness design, began reconciling current requirements to the legacy document
04/10/07	Matt Denko	Draft	Started initial draft

1.2 Team Members

Role	Name	Date of Review and Sign-off
Product Design	Matt Denko	
Product Management	Derry Vaughan, Aaron Taylor	
Project Management	Diane Lee-Dodele	
SWD Development	Jorge Lopez	
SWD Management	Max Deardorff	
TD	Mangesh Kalbhor	
TD Management	Partha Mitra	
User Assistance	Eileen Foran	
Additional contributors		

1.3 Related Documents

File Name	Description
\\acad\projects\Process and Power\Cork\Product Management\Cork MRD - First Draft.doc>	Cork MRD
\\acad\projects\Process and Power\Cork\Product Management\Cork Requirements – Validation Rev 1.doc	Validation requirements document

File Name	Description

2 Introduction

Standards are an integral component in the creation of meaningful project documents across a variety of industries. Applications can take different approaches to ensure standard compliance, ranging from rigid enforcement to assisted notification. An example of the former would be an application that ships with a predefined set of industry layers, and does not allow the user to create or modify this default set. An example of the latter is the standards utility included with vanilla AutoCAD, which provides a system tray notification mechanism that alerts the user when the drawing is not standards compliant as well as a standards checking utility which reports detailed information on each standards violation.

Standards are an important part of P&ID documentation as well. The Cork application's project management environment makes it easy for drafters to ensure standards compliance by enabling a Tool Palette work-flow that only exposes the discrete set of components appropriate for the current project context. Cork also strives to strike a balance, however, and has the flexibility to allow users to break standards by creating AutoCAD primitives or introducing external components that are not part of the core project. To address these cases, Cork includes a Validation feature that audits one or more drawings from the current project and reports on the following potential sources of error:

- Size or other property mismatches between in-line components and their associated lines
- Lines that don't terminate
- Conflicts in flow direction
- Unmated off-page connectors
- Incomplete tags
- Non-annotated lines
- AutoCAD primitives (annotations, lines, etc.) and blocks

2.1 Terminology

Term	Definition
Annotation	A general term for text-based AutoCAD primitives. Examples of annotations are text, dtext, mtext, and dimension objects. For the purposes of this document, AutoCAD-based annotations are distinctly different from native Cork *tags*.
Attribute	An optional data element that can be associated with an AutoCAD block. For example, a block representing a valve might have associated attributes representing its type and size. Attributes can also expose an optional annotation element that is placed with individual block instances.
AutoCAD Primitive	Native AutoCAD objects such as lines, arcs, and circles.
Block	Compound objects built up from a collection of AutoCAD primitives. Blocks provide a convenient mechanism for users to aggregate individual elements into a group which can be inserted as a single unit, saving the user from having

Term	Definition
	to manually redraw the individual elements that constitute it.
Component	Native Cork object types. Cork exposes a number of different object types, including the following: • Lines • Equipment • Valves • Fittings • Instruments • Non-engineering Items
Standard	An accepted convention that details best practices for a given industry in a particular region. For working with a P&ID, a standard might define components such as agreed upon symbols for representing different component types using blocks, as well as layering conventions, linetype conventions, etc. Applications take different approaches with regards to standards, ranging from rigid enforcement to complete drafting freedom. Cork strives for a middle ground, allowing users to draft as they are accustomed to while providing notification when non-standard or problematic situations arise and also supporting auditing for a complete accounting of all problem states across a given project.
Tag	An annotation element that displays property information about a Cork component.
Tool Palette	An AutoCAD UI element for grouping tools. In Cork, different projects populate the tool palette with varying tools. CAD Managers will typically define the available components, standard linetypes, etc. that are available for a given project when creating the project. This process populates the palette with appropriate set of project-based tools.

3 Goals, Requirements and Use Cases

3.1 Problems and Opportunities

Cork is Autodesk's second P&ID release in the Plant design space. A key differentiator for Autodesk versus existing applications in the market is usability. Anything our P&ID application can do to ensure accuracy in customer drawings in an automated or semi-automated manner will elevate Cork from the other offerings on the market.

3.2 Usability Goals

The primary usability goals for the P&ID feature are:

1. Support an interaction model that does not interrupt the user's work flow when they introduce potential errors into their P&ID drawings. The application should assume intelligence on the part of the user, and should at all times allow users to work as they see fit, even when potentially introducing problems into a drawing.
2. Provide sufficient visual cues and notifications within the application environment so that users can readily identify when a DWG is in a potential problem state.
3. Integrate P&ID Validation into key work-flow processes (e.g., by default the drawing should be validated prior to plotting, etc.)
4. Allow users to configure the application so that it doesn't check for certain problem states.

3.3 Marketing & Customer Requirements

The following requirements are copied verbatim from the revision of the Marketing requirements document that was available when this CD was created. Please note that in all cases, the actual marketing document should be the definitive reference for Cork scope and requirements.

3.3.1 Priority 1 Requirements

- Validation checks will need to be performed from within an AutoCAD P&ID project. The user who has access to the project shall be able to select one or more drawings on which to perform validation checks.
- After executing a validation check, the user shall have the option to save an electronic copy of the drawing or generate a plot of the drawing(s) showing all of the highlighted problem entities and the details of the error conditions for each.
- Each problem entity found during a validation check shall be visually different from the correct entities in the drawing. The user shall be able to look at the drawing(s) electronically or on paper plots and easily locate the problem entities.
- The user shall be able to see the details of the error conditions for each problem entity causing the validation problem on the drawing. When displaying the problem details, the user shall see both the current and preferred setting for that property. If more than one property is causing an error condition then all of the error properties shall be listed.
- The user shall be able to select a problem entity and remove it from the next validation check executed on the drawing. Users shall be able to select an 'ignored' problem entity and have it re-included in the next validation check.

- Cork shall check for and identify unmatched off-page connectors which are not connected to another off-page connector.

- Cork shall check for AutoCAD primitives and AutoCAD blocks that have been placed on a drawing but are not part of the project symbol library

- Cork shall check for inconsistencies between the properties of an asset which have been propagated from another asset (e.g., the size, spec, area number, system code, etc. of an inline asset like a valve which are propagated from the process line)

- Cork should check for all AutoCAD P&ID lines which have not been terminated at another asset, process line, or non-engineering item

- Cork shall check for conflicts between inline assets which have a flow direction property and the flow direction of the line on which they are connected

- Pipelines in a P&ID drawing should be annotated. Cork shall check to see if each of the pipelines have been correctly annotated based on the setup defined for the project.

- The Validation feature shall make use of or be designed similar to any existing AutoCAD features where possible.

- The user shall be able to see a report of all validation errors on a drawing or across a project. When viewing validation errors for a project, the user shall be able to group the errors per drawing number.

- When viewing a report of the validation errors, the user shall be able to select a validation error and have the program automatically zoom to and highlight the selected validation error.

3.3.2 Priority 2 Requirements

- Cork shall check for conflicts in the flow directions of multiple lines meeting at a 'T' or branch connection. These conflicts shall include flow conditions like the flow direction of all three lines being towards the T and the flow direction of all three lines being away from the T.

- For all assets with a tag number property, Cork shall check to ensure that all of the variables of the tag number have been completely filled.

- While viewing the drawing file records of drawings in a project, the user shall be able to see a visual notification that a particular drawing file has validation errors.

- Tag numbering formats and annotation style definitions may change during a project. Cork shall check for discrepancies between the annotations on a drawing and their respective annotation structures as defined in the project setup.

- For complicated drawings and large projects the execution of a validation check may take some minutes. The user shall receive visual feedback as to the status or progress of the validation check.

- The user shall be allowed to fix a validation problem manually and then re-run the validation command. There is no requirement to automatically fix a validation error.

- As there are many different validation checks which could be run on a drawing, a user should have the ability to define their own custom validation check routines. The user should have the ability to create their own validation checks using a programming language like LIS, VB, etc. and include them as part of the execution of the product Validation checks.

- Prior to generating a report or plot of a drawing, the user shall be allowed to have the drawing checked for validation errors.

3.3.3 Priority 3 Requirements

None identified.

3.3.4 Usability Requirements

None identified.

3.4 Use Cases

Use cases are contained in the MRD: \\acad\projects\Process and Power\Cork\Product Management\Cork MRD - First Draft.doc

4 Feature Design

4.1 Conceptual Models

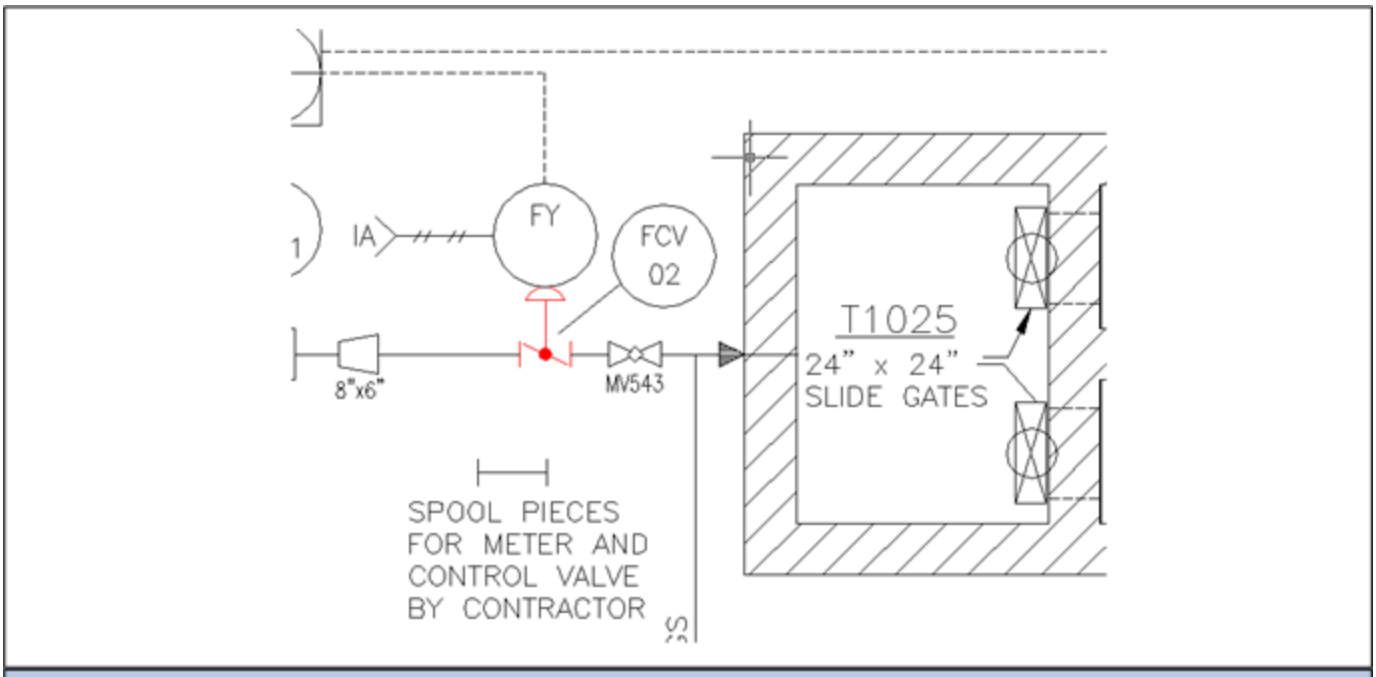
4.1.1 Overview

Cork supports a project-based work-flow[1]. A CAD Manager will typically setup a project, identifying the appropriate components (lines, pumps, valves, tanks, etc.) that drafters will work with. Drafters will create individual P&IDs largely by dragging components from AutoCAD Tool Palettes and dropping them into a drawing. Throughout the drafting process, users can potentially introduce errors in a number of ways, most notably the following:

- **By placing an AutoCAD primitive or block into the drawing, rather than placing a Cork component via the project Tool Palette.** AutoCAD objects have no direct linkage to the Cork context, and thus will not show up in reports. This can be particularly problematic in cases where the AutoCAD object closely resembles a Cork object – for example, inserting an AutoCAD block into a drawing rather than a Cork recognized component. If a symbol with similar or identical appearance exists in the current project, consumers of hard-copy printouts of the P&ID will potentially interpret the AutoCAD block as the Cork component, but any reports generated for the project will not include the AutoCAD block[2].

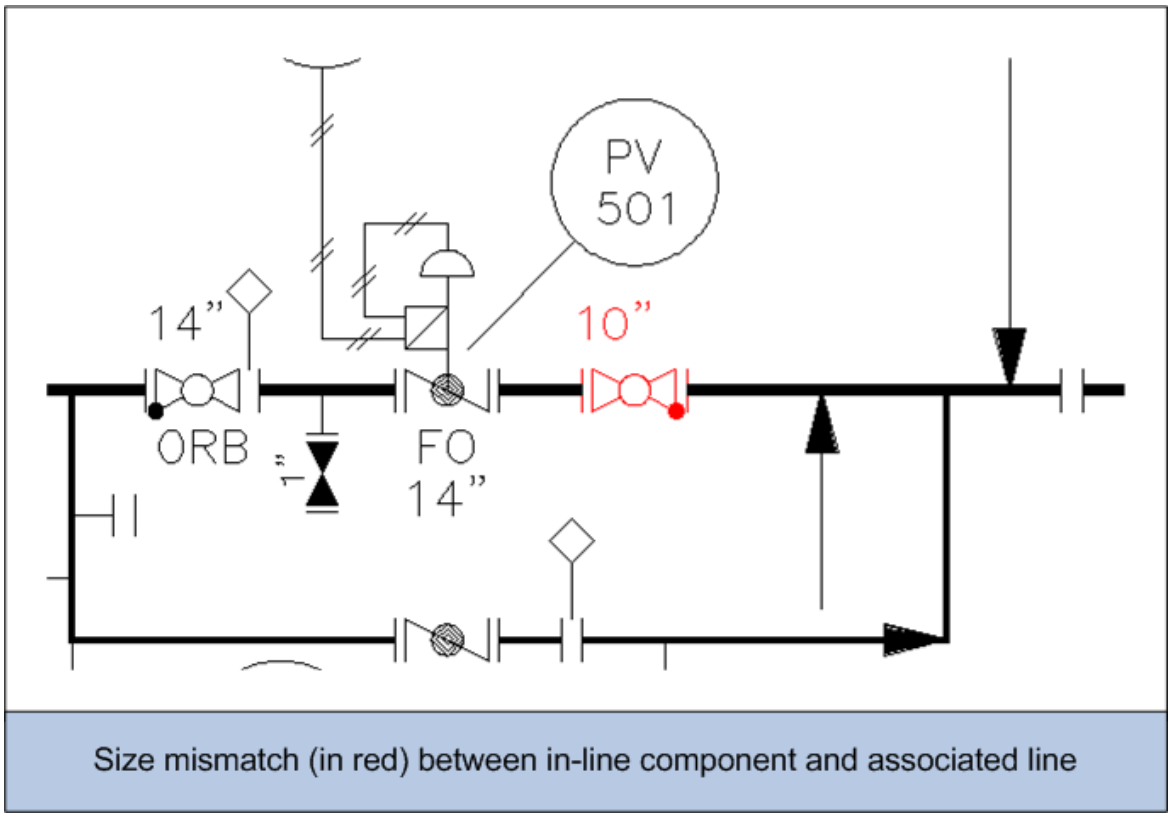
[1] Details of the Guinness work-flow are largely detailed in other specifications (refer to the Related Documents section of this specification for a complete accounting). Please refer to the appropriate spec for additional information.

[2] Note: if users classify an AutoCAD object, it is not reported as part of the validation process, and the object is captured as an error in the Project Manager. Similarly, any native Cork objects which are actually block objects are not reported as errors.



AutoCAD block insert (in red): looks like P&ID asset but isn't captured in reports

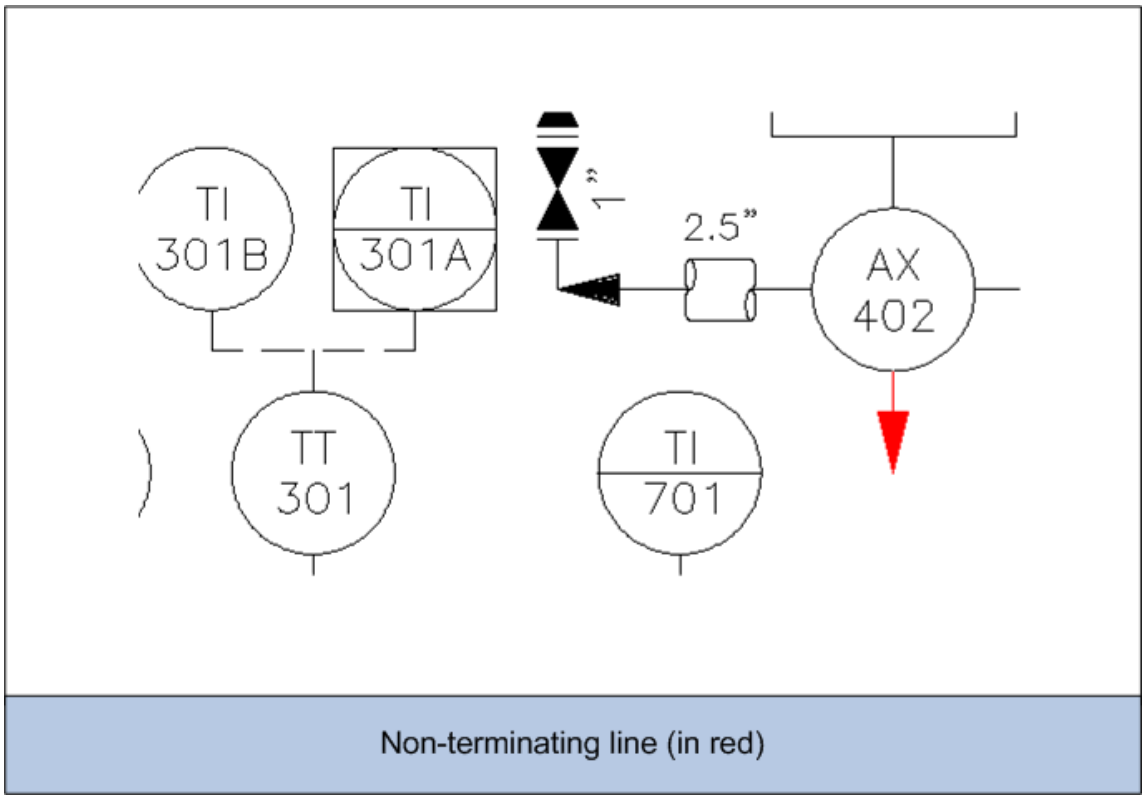
- **Property mismatches between in-line components and their associated schematic lines.** By default, in-line components inherit a number of properties from the line they are associated with. If users subsequently change a line property such as size, the in-line component updates itself to reflect the new size. This linkage is not bidirectional, however – if users update the size of the in-line component through the Properties Palette, a mismatch can occur between the component and the line.



Size mismatch (in red) between in-line component and associated line

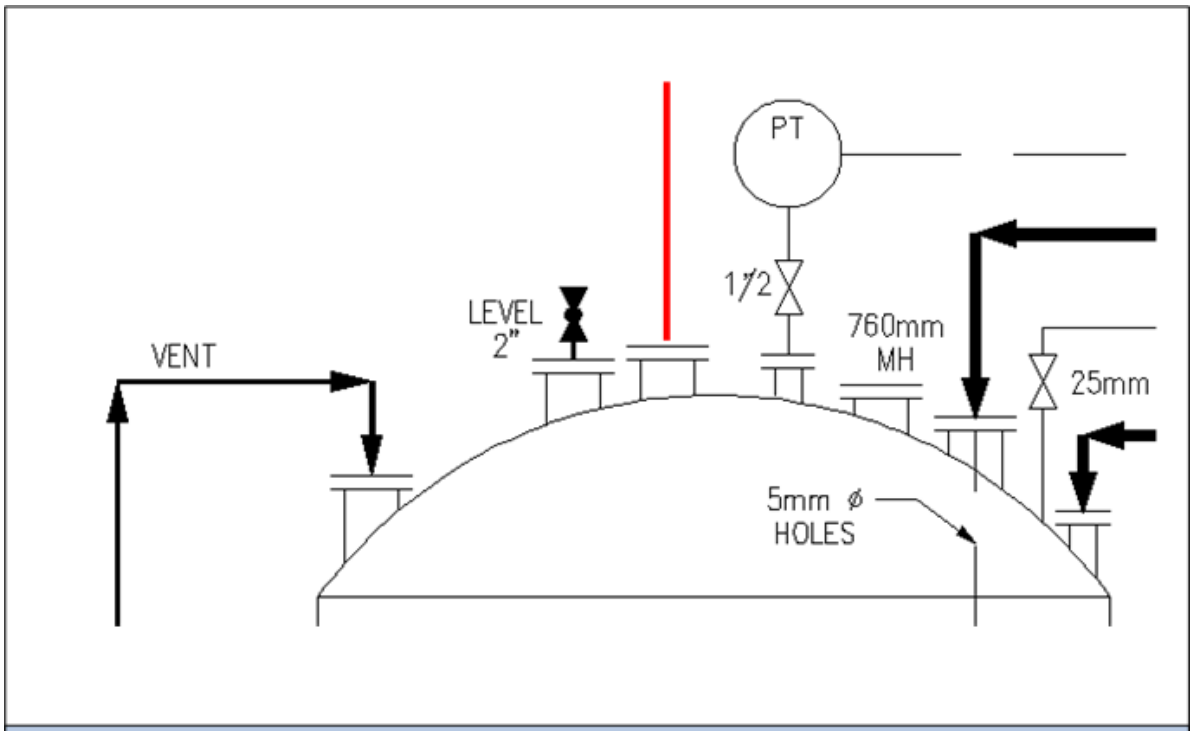
- **Schematic lines that don't terminate.** Cork project lines should terminate either through a connection to an end-line component or an end of line symbol with associated annotation indicating what the termination means (e.g., air vent, water outlet, drain, etc.). Users may inadvertently neglect to terminate one or more lines in a project drawing, and these by default are flagged as errors.
- **Process lines that aren't annotated.** All process lines in a P&ID should be assigned a tag and properly annotated in the P&ID. Any process line[3] that isn't should be flagged as a validation error.
- **Incomplete tag data.** Tags are composed of a number of fields. If any of the fields in a given tag are blank, a validation error is returned.

[3] Note: process line refers to any schematic line that is exposed through the Pipe Line grouping in the Tool Palette. Any schematic line that appears in the Instrument Lines grouping is not considered a process line, and is thus not subject to validation checking.



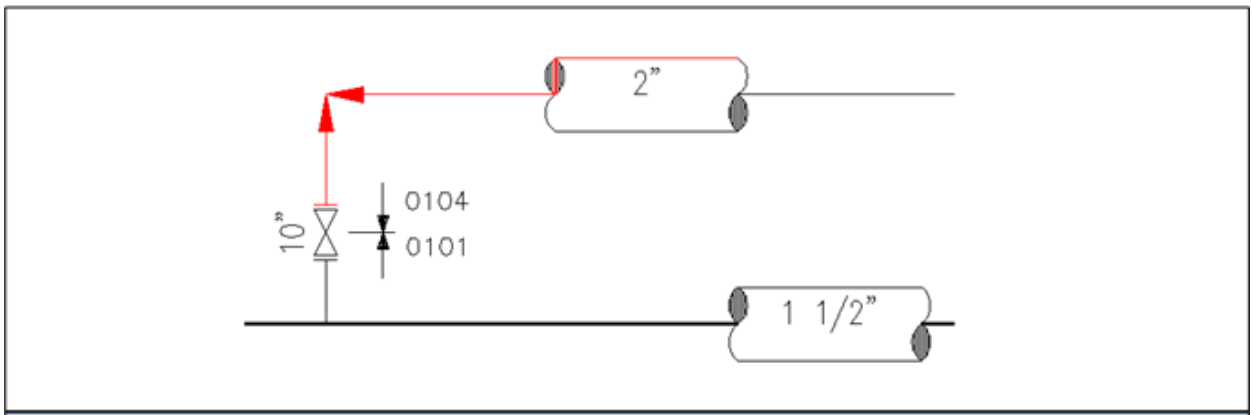
Non-terminating line (in red)

- **Off-page connectors that don't resolve.** Off -page arrows are used in P&ID drawings to indicate lines that are continued in another drawing file. It is typical to place off-page connectors initially without pointing them to another drawing file, largely because the drafter isn't sure which file to use or because the other file might not yet exist. Once the project has been completed, however, it is important that all off-page connectors properly resolve.
- **In-line or end-line components that appear to be associated with a line but which aren't programmatically connected**. When creating drawings, users can either fail to properly establish a connection between a component and a line when initially placing the component, or after initial placement by dragging the component away from the line.



Line which appears to be connected (in red) to tank but isn't

- **Conflicts in flow direction.** All lines have a flow property (as do some in-line components such as valves), which indicates the direction of movement for the material conveyed by the line. Flow direction can optionally be depicted symbolically in P&ID drawings by an arrow, but oftentimes flow is not graphically depicted. In these cases, it may be difficult for users to identify flow conflicts without an auditing tool.



Flow direction conflict (indicated in red)

- **Duplicate tag numbers for a given object type.** In most cases it is desirable (if not required) to assign unique tag numbers to objects of a given type, such as pumps. Duplicate tag numbers can lead to confusion, and make it impossible to unambiguously identify a component.

When any of these situations occur in a project drawing, it is important that Cork notify users that problems exist and provide a mechanism for reviewing and fixing them. This is handled in the following ways:

- **Automatic notification of problem states when users attempt to perform certain operations on a project drawing that has validation errors.** When users perform these operations from within the project context, an alert is displayed which notifies users that the drawing has validation problems. Users can choose to ignore the alert and continue with the operation, or to review the problems in a secondary UI. By default, a validation operation will automatically be performed when users attempt to plot or publish a Cork project drawing.
- **User-initiated validation operation.** Users can also explicitly invoke a validation operation on one or more project drawings by creating a selection set in the Project Manager, then right-clicking and choosing the Validate option.
- **Background polling of project drawings.** Users will be able to configure AutoCAD so that validation checks are performed in the background at specified polling intervals. When configured for this mode of operation, a tray icon will be exposed at the bottom of the AutoCAD drawing editor. If any validation errors are encountered during a polling interval, a bubble notification will be displayed alerting users to the problem.

4.1.1.1 Default Workflow

The default, out-of the box configuration of the Cork Validation feature will enable background polling of project drawings for validation errors. When a Cork project is open in the AutoCAD drawing editor, a validation operation will run as a background process at a user-settable interval (initially set to every fifteen minutes). If any validation errors are encountered in any project drawings, an icon indicator is presented in the Project Manager tree to visually differentiate the problem drawings[4]. A bubble notification is also displayed in the system tray to alert users to the problem.

The Project Manager will introduce a new Validation Errors panel, from which users can traverse a set of validation errors in individual drawings selected from the tree, or all errors in the project by selecting the project node. Each error will also be visually indicated in the drawing editor via a markup with an associated annotation describing the encountered error condition, as depicted in the following screenshot:

[4] Each project drawing will also have a dedicated Dashboard panel that lists any validation errors, as described later in this document.

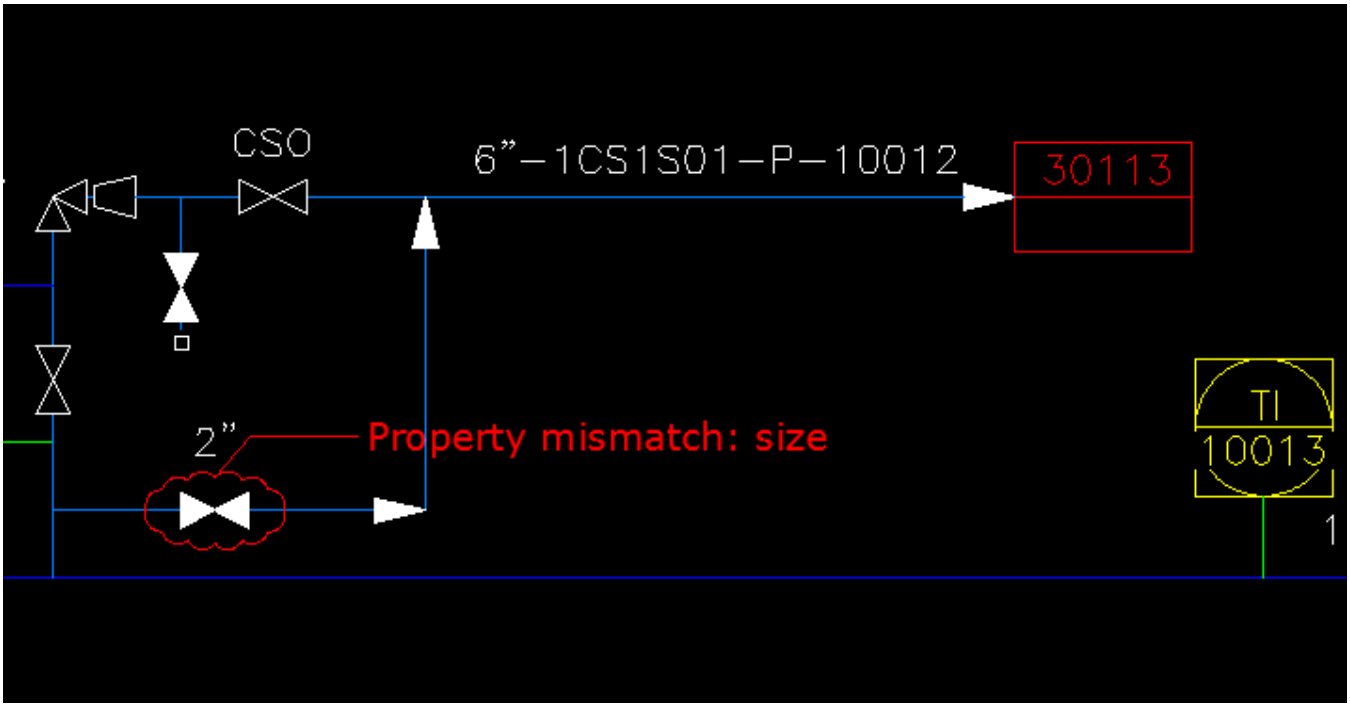


Figure 1: Validation Markup

The markup element is currently envisioned as a pseudo object type. As opposed to true AutoCAD objects, the validation markup will not be directly user selectable. To be more specific, users will not be able to click on the validation markup to perform operations on it (for example, in order to move or delete it). When users mouse over the markup object, an enhanced tool-tip will be displayed, leveraging the Raptor UI enhancements being undertaken by the AutoCAD team. This tool-tip will provide additional information about the error.

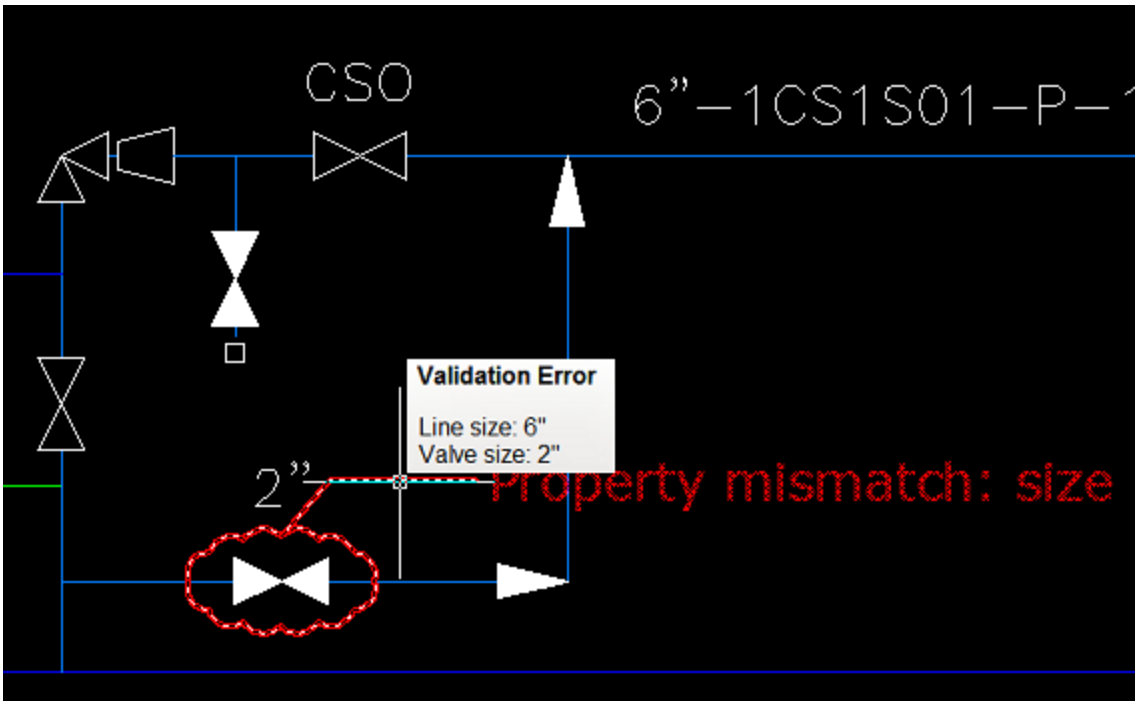


Figure 2: Validation Error Tool-tip

By leveraging the progressive disclosure element of the enhanced tool-tips, more detailed User Assistance Help content can also be provided for individual error types if the user continues to hover the cursor over the validation markup object.

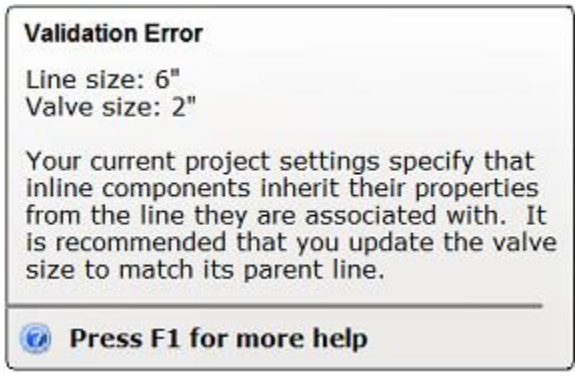
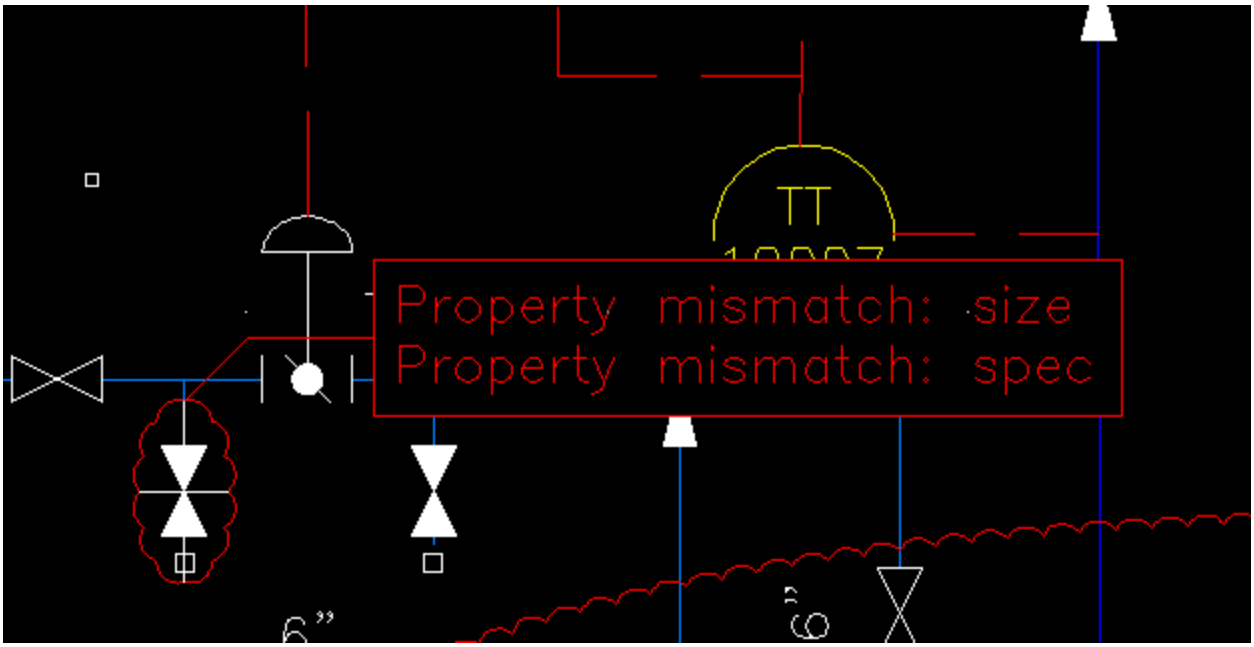


Figure 3: Validation Tool-tip Progressive Disclosure

If multiple validation errors exist for a given object, each subsequent validation error is presented on a separate in a multi-line label. As users hover over individual lines, a tool-tip is displayed as previously described that provides specifics about the current error line. As an example, consider a situation where an inline component had two inherited property values that had been edited and no longer matched the schematic line the component was associated with. In this case, a multi-line tool-tip such as the following would be displayed:



Note that by default the markup label applies a wipeout to objects below it to make the error label more legible, but users can toggle off the markup label if desired as described later in this specification.

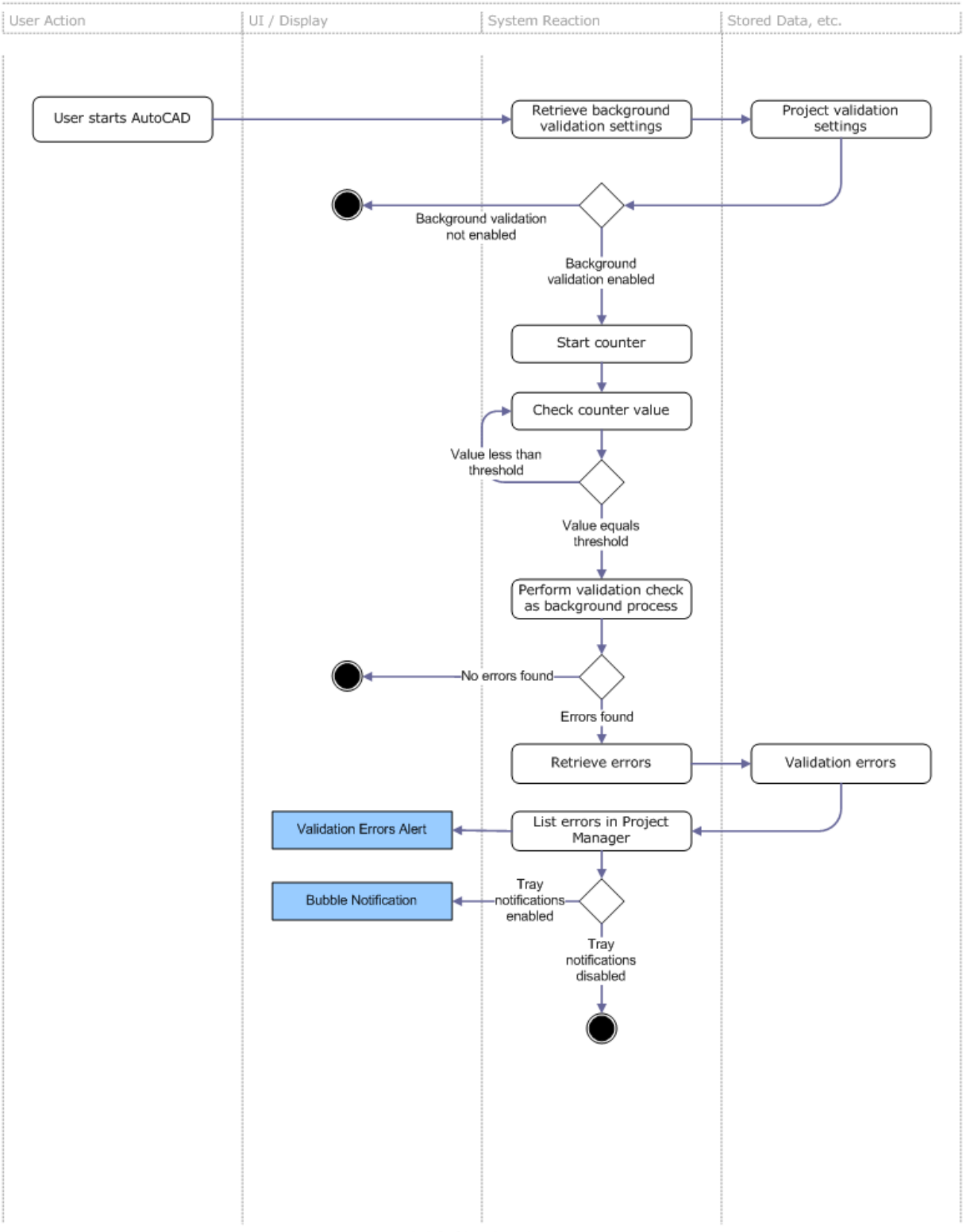
Once a Cork object has triggered a validation error, it will have a reactor associated with it that monitors the object for editing operations. When edits are performed, the object will check to see if the operation fixes any associated validation error conditions. Once all error states are resolved on the object, the reactor is removed, and the associated validation markup and annotation are removed from the drawing[5]. Users can also suppress the display of an individual validation markup by marking the error as ignored as described later in this document.

In addition to the background polling of project drawings for validation errors, users can configure Cork to perform validation operations when certain operations are performed, or can explicitly invoke a validation operation on one or more project drawings at any time. Any validation errors encountered will be listed in the Project Manager interface as previously described.

The following action diagram depicts the primary validation work flow.

[5] The error record is also removed from the Project Manager and Dashboard interfaces.

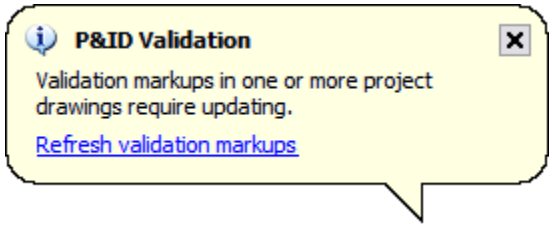
AD 01: Background Validation



4.1.1.1.1 Validation Tray Notifications

If users have enabled background polling for validation errors, a validation check will be performed on any drawings from the current project that the user currently has opened for editing. The check will search for any ***new*** validation errors that have been introduced over the course of the editing session. The reason the check is performed for new errors is to determine if the current display of validation markups in the collection of working drawings is up-to-date. If any new errors have been introduced, a balloon notification is presented to the user in the system tray, notifying that the additional validation errors have occurred and that the validation markups should be refreshed to reflect these additional errors. The validation check does not look for cases where a validation error has been fixed, as this case results in the instantaneous removal of the validation markup.

The following screenshot depicts the Validation tray notification:



A tray icon (to be developed later in the Cork cycle) is present in the system tray at all times if background validation is enabled. If users click on the icon, the P&ID Validation Settings dialog box is invoked. If users right-click the tray icon, a context menu with the following options is exposed (accelerators indicated as underlined text):

- **Configure Validation Settings**… - Launches the P&ID Validation Settings dialog box.
- **Validate Current Drawing** – Performs a validation check on the current drawing. This option is disabled if the current drawing is not a member of the current project, or if no project is loaded in the Project Manager.
- **Validate Current Project** – Performs a validation check of all drawings in the current project. This option is disabled if no project is loaded in the Project Manager.
- **Refresh Validation Markups** – Refreshes the validation markups of any currently open project drawings whose markups are out of synch. This option is disabled if no new validation errors have been identified via a background validation check.

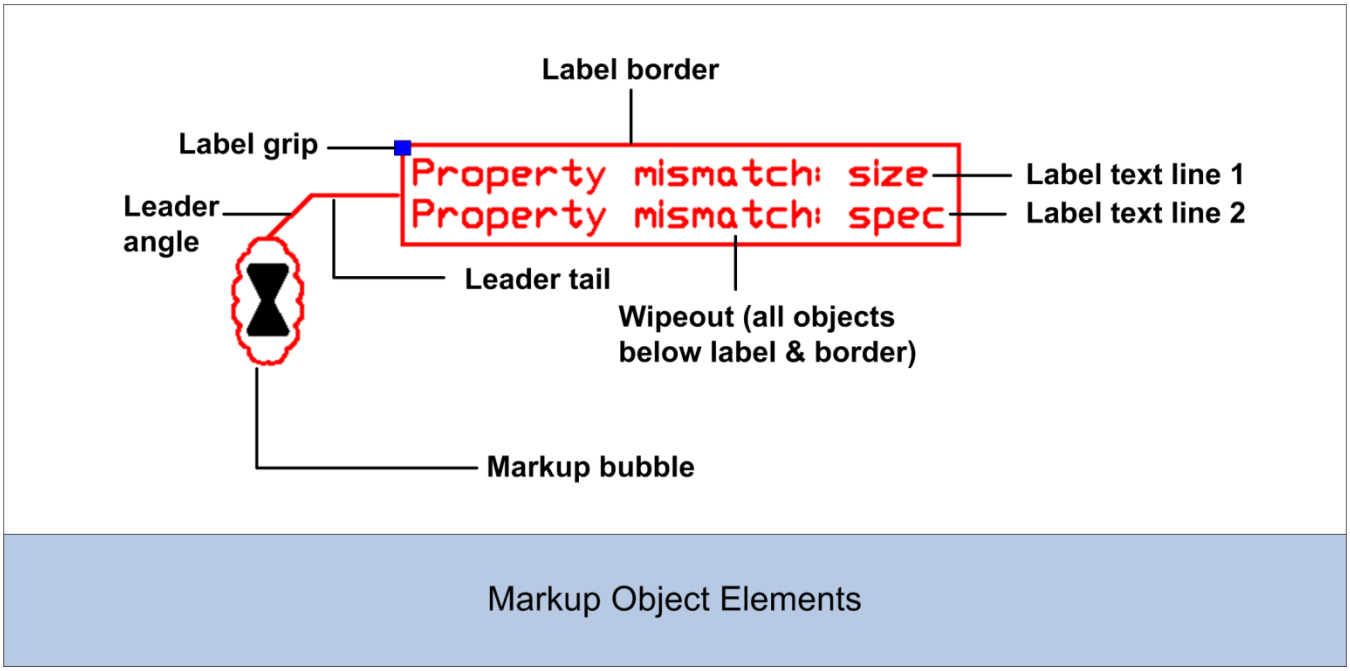
The Validation tray icon will honor the current global user settings for tray icons. More specifically, if users have disabled tray icons or tray notifications, no icon or notification will be displayed, regardless of the current setting in the P&ID Validation Settings dialog box. Similarly, if notifications are enabled, the system setting for how long the notification is displayed (until dismissed by the user or for a discrete period of time) will be used to determine the behavior of the notification.

4.1.1.1.2 Validation Error Markups

Each validation error has an associated markup which is displayed by default in the drawing editor. These Markup objects are comprised of a number of discrete elements:

- **Markup bubble** – a balloon-like element which is drawn around the object that triggered the validation error.
- **Leader angle** – an angled leader element which is drawn by default at a 45 degree angle.
- **Leader tail** – a horizontal leader element which is connected to the end of the leader angle.
- **Label border** – a rectangular bounding box which is drawn around the text of the markup label.

- **Label text** – the text that indicates the validation error or errors that is associated with the object. If multiple errors are present, each individual error is represented on a separate line.
- **Label wipeout** – an optional element that masks everything below the label border and text, to make it easier for users to read the markup label.
- **Label grip** – a grip element that allows users to reposition the label border and text. As these elements are repositioned, the leader element redraws itself to maintain a connection to the label.



Markups are pseudo object types, in that they cannot be directly manipulated by users. More specifically, users cannot select a markup object and manipulate it (move it, delete it, change properties via the Properties Palette, etc.). In actuality, the preceding statement isn't strictly true as grips are exposed for repositioning the markup label text, but in practice users are largely restricted in their direct interactions with the markup object.

ISSUE: May want to kill the grip and just expose a right-click menu option that allows users to reposition the label. It might be conceptually confusing to allow users to select the label, but not be able to perform standard AutoCAD operations on. Discuss during feature team review.

All markup elements share a common color, and all elements except label text share a common lineweight[6]. Different default values are specified in imperial versus metric drawings. In the following sections, each element will be described in greater detail.

Markup Update Behavior

Markups are updated due to the following actions:

[6] Note: SHX fonts also share a common lineweight.

- **Saving a project drawing with inconsistent markup display**. Background validation operations may uncover new validation errors during an editing session, but if users don't manually refresh markups via the system tray the markup data in the erroneous drawing will be stale. Similarly, users may have tray notifications disabled, in which case they will not receive an explicit notification or the opportunity to refresh markups. To address these situations, when users perform a save operation on any project drawing with inconsistent markup display, markups are refreshed as part of the save operation.

- **Opening a project drawing with inconsistent markup display.** Users may perform explicit validation operations (either by typing VALIDATE at the command line or by choosing the Validate option from the Project Manager). In the event these operations uncover new validation errors, the project needs to track which drawings have stale (i.e., incomplete) markups. The next time such drawings are opened by a user from the project, the markup data is refreshed.

- **Users perform an action that fixes a validation error**. When users fix a validation error, the following activities take place immediately:

    - The corresponding validation error row in the Project Manager is removed

    - The validation markup is removed from the associated object

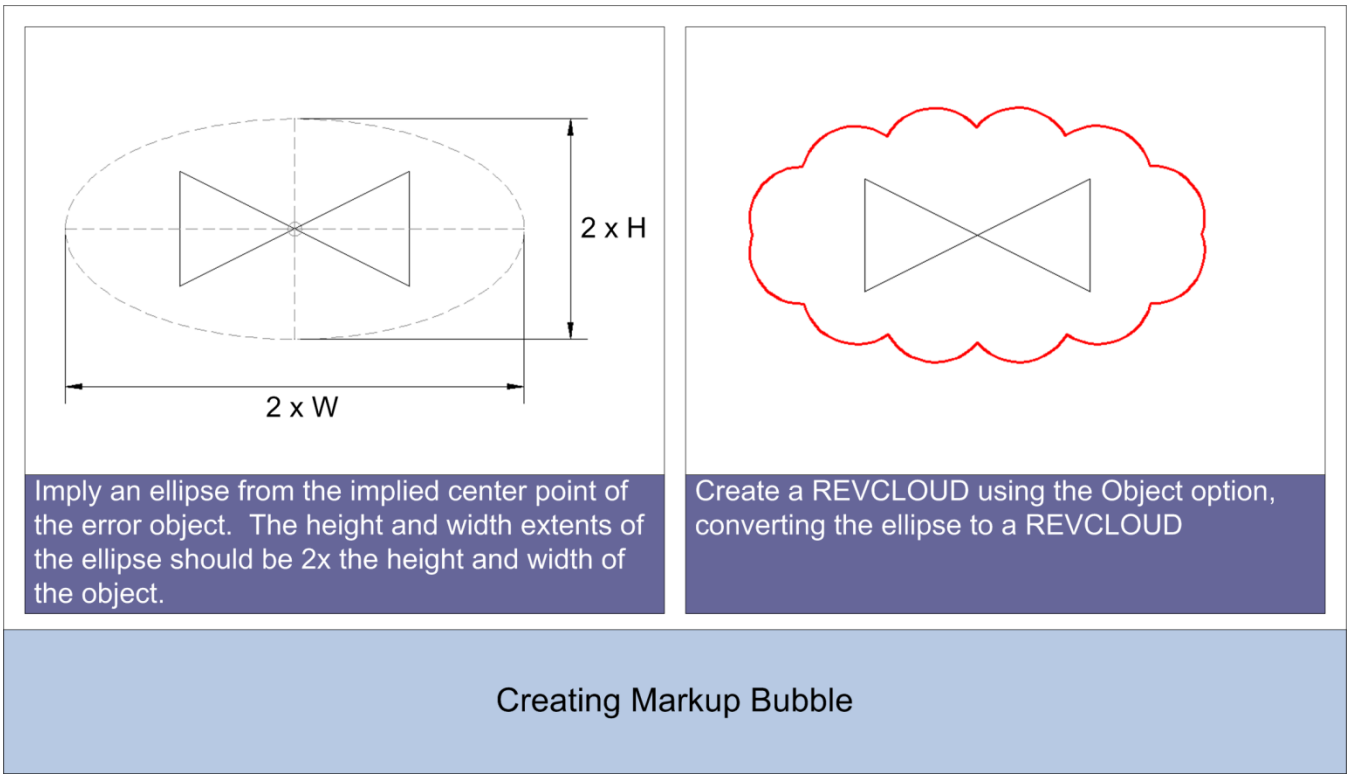
  If users perform an UNDO operation that reverts the error condition, the validation error row and validation markup must be added back.

- **Users manually refresh markup display via tray notification**. When this occurs, any project drawings that are currently open with stale markup data are refreshed.

Markup Bubble

Each object that has a validation error associated with it has a markup bubble drawn around the erroneous object. While solely at the discretion of SWD to determine how best to implement the markup bubble, design envisions behavior similar to a work-flow that can be created using standard AutoCAD objects:

- An ellipse is drawn around the error object in the following manner:
    - The center point of the error object is identified
    - A temporary line measuring two times the width of the error object, with its midpoint located at the object center point, is drawn
    - A temporary line measuring two times the height of the error object, with its midpoint located at the object center point, is drawn
    - An ellipse is created using the Center option to locate the center of the ellipse so it is located at the center point of the error object, with the end points of the ellipse located at one of the end points of the width and height temporary lines
- The REVCLOUD command is used to create a revision cloud, using the Object option to select the ellipse created in the preceding steps

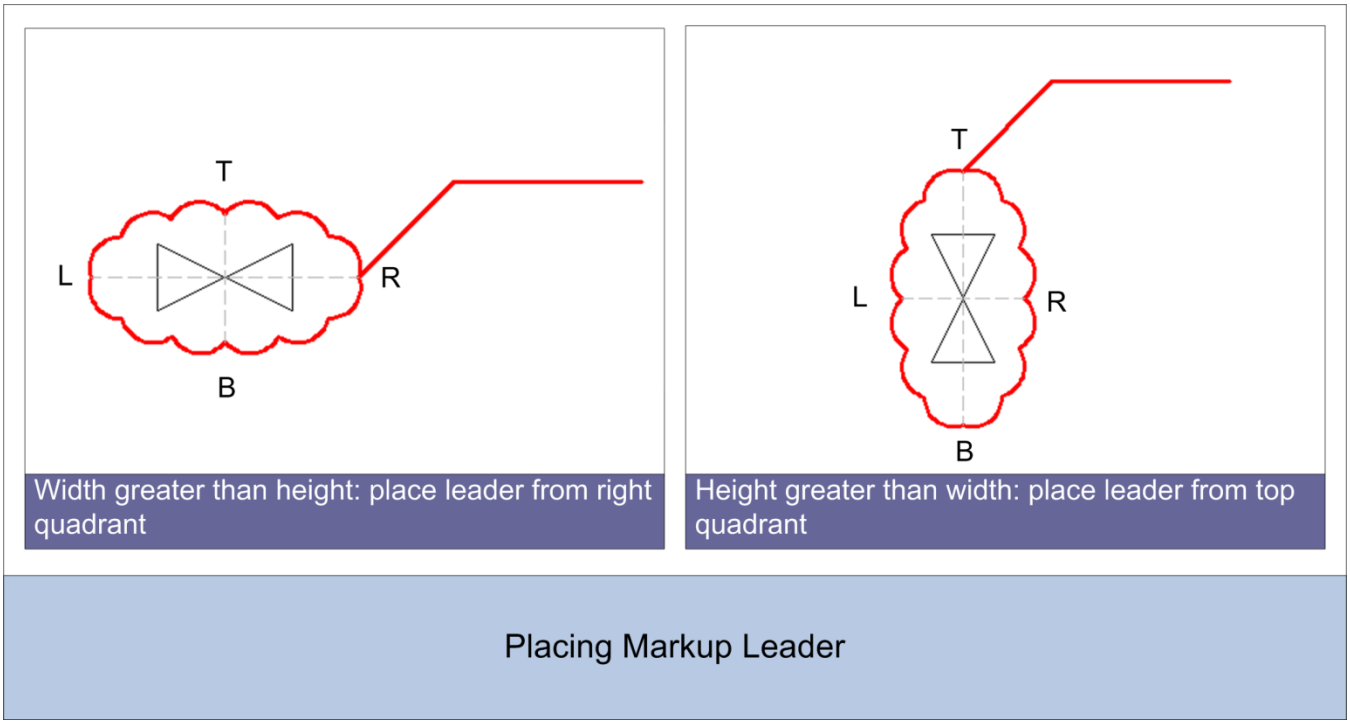


Markup Leader

The markup leader is comprised of an angle and a tail portion. The angle leader is drawn at a 45 degree angle from the largest of the following markup bubble quadrants:

- Top
- Right

More specifically, if the bubble is taller than it is wide, the leader tail is drawn from the upper quadrant; if the bubble is wider than tall, the leader is drawn from the right quadrant.



Placing Markup Leader

Specifics about the property values assigned to the leader elements are described in the Imperial vs. Metric Properties discussion later in this section.

Markup Label

The markup label is comprised of a text element, a border that is placed around the text, and an optional wipeout element.

Markup Label Margins: Consistent Padding Between Label and Border

Imperial vs. Metric Properties

Certain properties of the markup bubble vary depending on whether the user is working from a metric vs. imperial drawing template, as detailed in the following table.

Property	Imperial	Metric
Markup Bubble Arc Length	.10	2.5
Leader Angle Length	.25	6.25
Leader Tail Length	.35	8.75
Label Border Margin	.03 on all sides	.75 on all sides
Lineweight of Markup Elements	.3 mm	.3 mm
Label Text Style	Standard	Standard

4.1.1.1.3 Validation Errors Summary

The following sections describe in greater detail the constitution of the various validation error types, and provide a high-level overview of the text that appears in the enhanced tool-tips when users hover over an object with a validation error in the drawing editor. While PD provides some general guidelines with respect to expected content in the extended (i.e., progressive disclosure) tool-tips, the expectation is that User Assistance will come up with the Help content that is presented in these Tool Tips. This content will be developed by UA after the spec review.

Property Mismatches

Property mismatches occur when users over-ride an inherited property value, either through the Properties Palette or through the Data Manager. For example, inline components by default inherit their size property from the schematic line they are associated with. If users change the size value after the initial placement of the component, a property mismatch may occur.

The Validation check should honor the current user settings with respect to what properties are inherited, and perform checks based on those settings. For more information, refer to the design of the Property Propagation feature.

If the current user's validation settings are configured to check for property mismatches, each property of inline or end-line components will be checked with the corresponding property of the schematic line they are associated with. If any inherited properties are mismatched, an error is returned. Similarly, any inherited property that has a null value should be considered as an error condition.

The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

Element	Text
Callout Label	Property mismatch: <*property_name*>
Tool-tip: Level 1 Header	Validation Error

Element	Text
Tool-tip: Level 1 Text	<*Property_source_object*> <*property_name*>: <*current_value*> <*Inherrited_property_object*> <*property_name*>: <*current_value*>[7]
Tool-tip: Level 2 Text	Your current project settings specify that inline and end-line components inherit their properties from the schematic line they're associated with. It is recommended that you update the component size to match its parent line[8].
Tool-tip: Footer	Press F1 for more Help

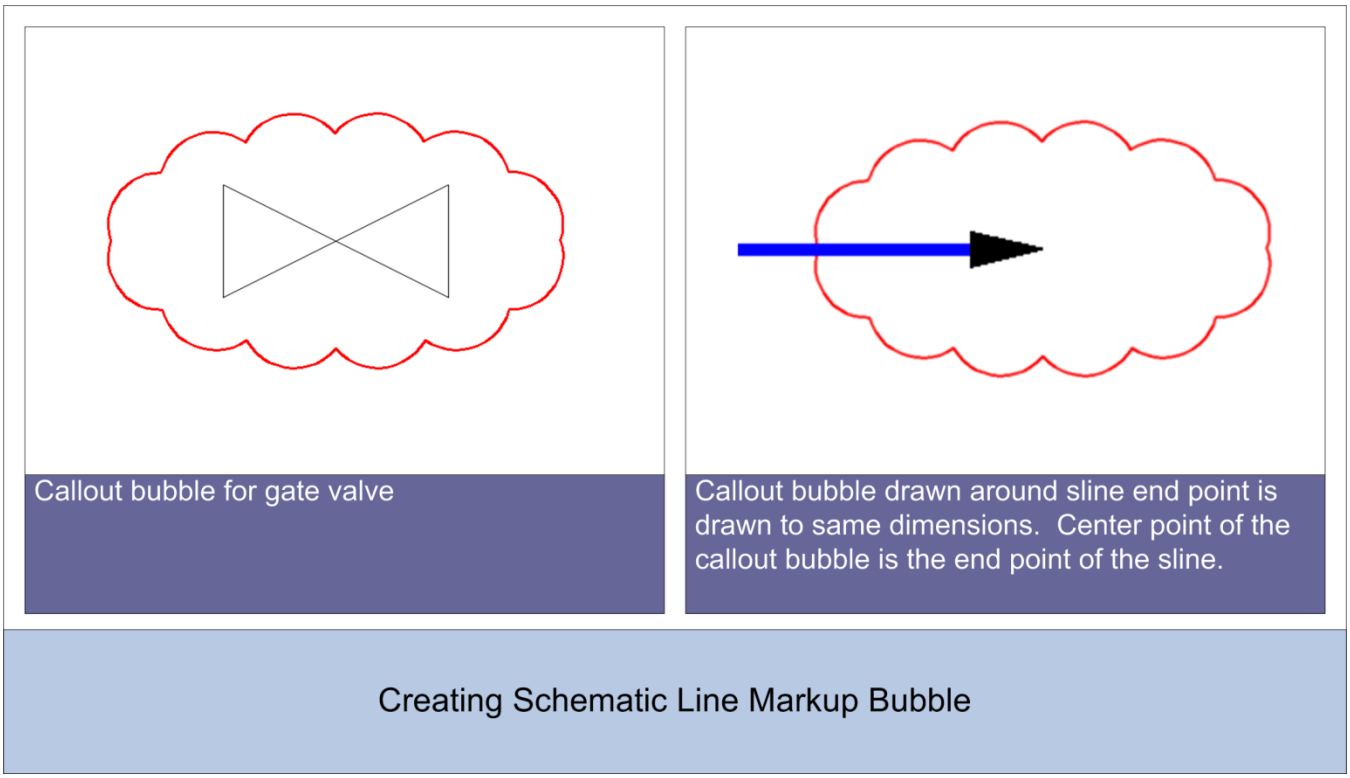
Non-terminating Lines

Any schematic line end point that doesn't terminate, either through an explicit connection to an end-line component or via a symbolic termination through an off-page connector, is considered a non-terminating line. When configured to check for this condition, a markup will be added to the end point of any non-connected schematic line.

The callout bubble should be drawn to the same dimensions as the bubble drawn around a gate valve. The end point of the sline should be assumed to be equivalent to the center point of the gate valve, as depicted in the following illustration.

[7] Where <*Property_source_object*> is the object from which the property value is inherited (for example, a schematic line), <*Inherited_property_object*> is the object inheriting the value (e.g., a valve), <*property_name*> is the name of the mismatched property (e.g., size), and <*current_value*> is the property value for each respective object.

[8] Exact verbiage to be determined by UA later in the Cork cycle



The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

Element	Text
Callout Label	Non-terminating line
Tool-tip: Level 1 Header	Validation Error
Tool-tip: Level 1 Text	Non-terminating line
Tool-tip: Level 2 Text	By convention, all schematic lines should terminate in a connection to another schematic line, a component, or an off-page connector.[9]
Tool-tip: Footer	Press F1 for more Help

Un-annotated Process Lines

All process lines must be properly annotated in the P&ID. If not, they trigger a validation error, with a callout bubble being placed at the midpoint of the schematic line drawn to the same dimensions as previously described for non-terminating line validation errors.

[9] Actual text to be delivered by UA

The following schematic lines are considered process lines and must be annotated in the drawing:

- Primary line segment
- Secondary line segment
- Primary line segment – new
- Secondary line segment – new
- Primary line segment – existing
- Secondary line segment – existing
- Jacketed pipe segment

The following schematic lines are not considered process lines, and should be excluded from any validation checks for this error condition:

- Electric signal
- Pneumatic signal
- Hydraulic signal
- Software link
- Capillary tube
- Instrument supply line
- Leader
- Electromagnetic signal
- Mechanical link

Note that this check only verifies that an annotation is associated with each process line, not that the tag is complete. If the annotation tag is missing data, that is returned as a separate incomplete tag error condition as described elsewhere in this section.

The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

Element	Text
Callout Label	Un-annotated Process Line
Tool-tip: Level 1 Header	Validation Error
Tool-tip: Level 1 Text	Un-annotated Process Line
Tool-tip: Level 2 Text	By convention, all process lines should be annotated in the P&ID. Any schematic line that appears in the Pipe Lines grouping in the Tool Palette is considered to be a process line. Any schematic line appearing in the Instrument Lines grouping in the Tool Palette is not considered to be a process line.[10]

[10] Actual text to be developed by UA later in the design cycle.

Element	Text
Tool-tip: Footer	Press F1 for more Help

Incomplete Tags

Tags are comprised of a number of discrete fields. If any of these fields are null, an incomplete tag error is returned. A callout bubble is drawn around the object with the incomplete tag. In the case of schematic lines, the callout bubble is drawn according to the rule previously described for un-annotated process lines.

The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

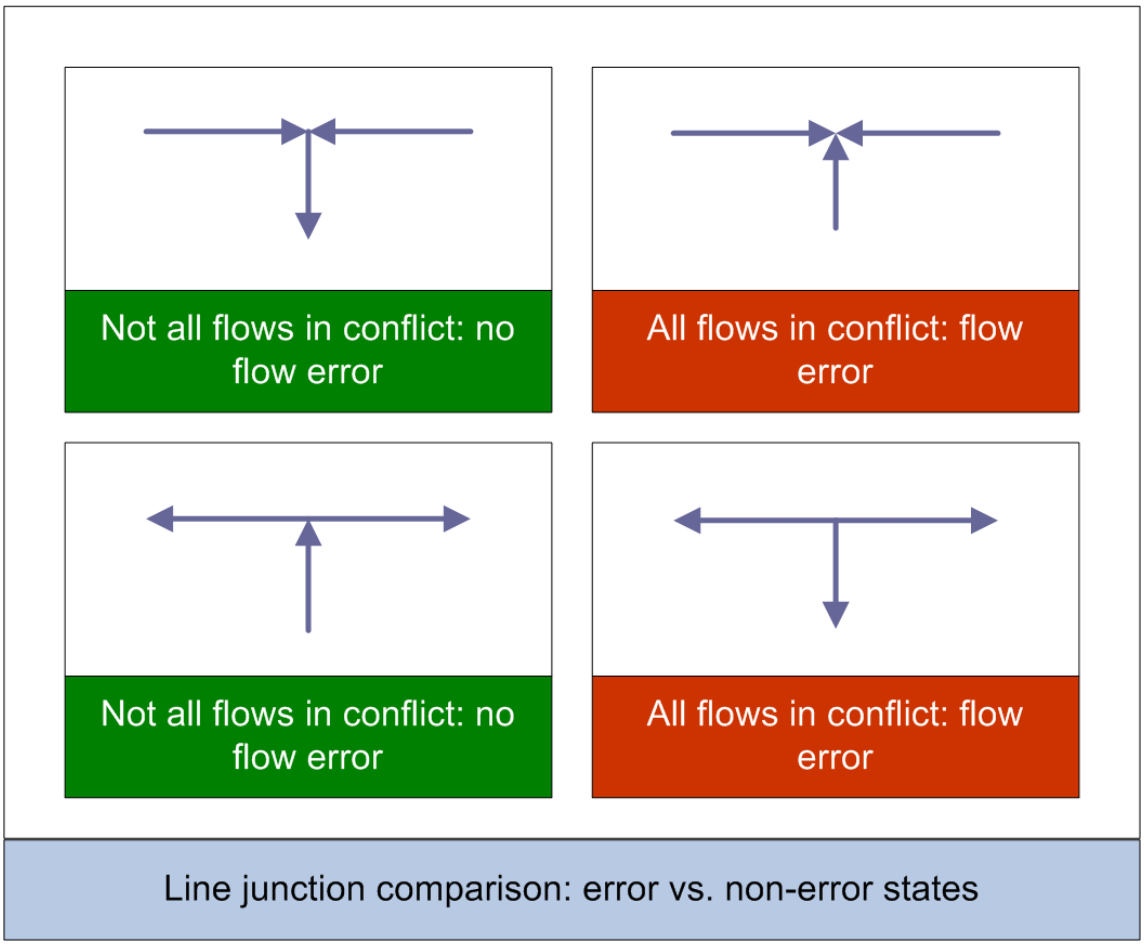
Element	Text
Callout Label	Incomplete Tag
Tool-tip: Level 1 Header	Validation Error
Tool-tip: Level 1 Text	Incomplete tag Missing data: <*field_1*>, <*field_2*>…<*field_n*>[11]
Tool-tip: Level 2 Text	Tags are composed of one or more field elements. If any of these fields are blank, the tag is considered incomplete.
Tool-tip: Footer	Press F1 for more Help

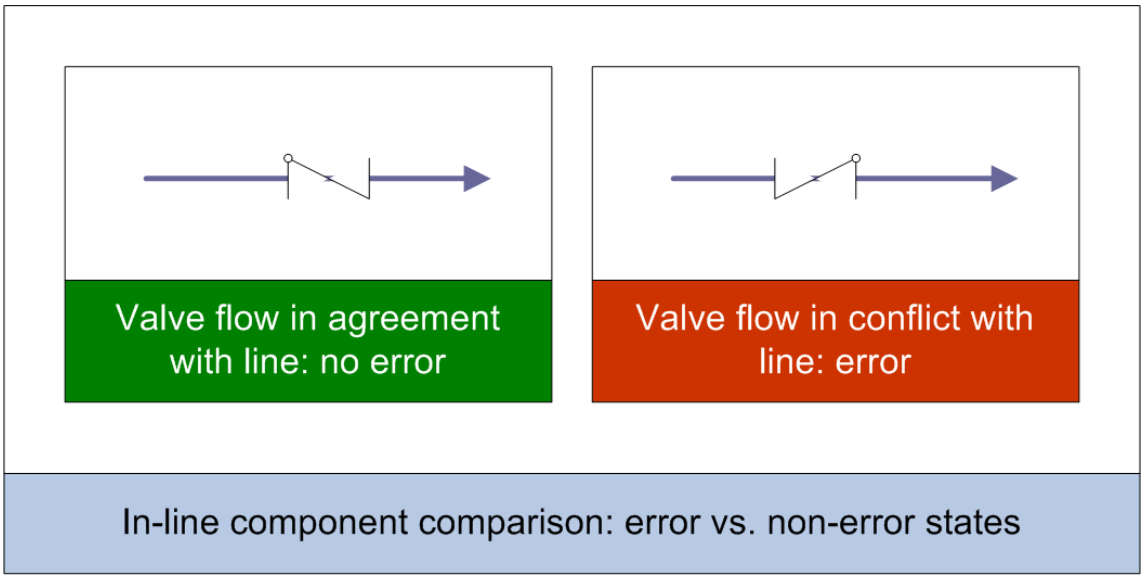
Flow Direction Conflicts

Flow direction errors can occur when users manually flip the flow direction of an element of a line segment. Two primary error types can occur:

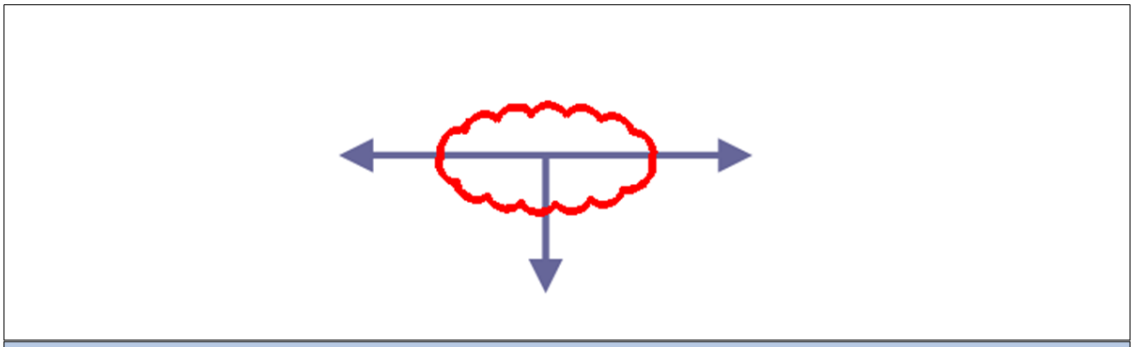
- Errors where all flows of a line segment at a connection junction are in conflict with each other
- Situations where an in-line component is in conflict with its associated line

[11] Where "<field_n> is equal to the last field of missing data for the erroneous tag





When flow direction conflicts occur, a markup is placed around the conflicting junction. For example, considering the final case from the previous illustration, a markup would be placed at the conflicting junction as follows:



Flow Conflict Markup

The markup is drawn to the same dimensions as a gate valve, with the conflicting intersection point serving as the center point of the bubble.

The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

Element	Text
Callout Label	Flow Conflict
Tool-tip: Level 1 Header	Validation Error
Tool-tip: Level 1 Text	Flow Conflict
Tool-tip: Level 2 Text	TBD
Tool-tip: Footer	Press F1 for more Help

Unconnected Components

All components should be connected to a schematic line. If not, a callout bubble is placed around the unconnected object.

The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

Element	Text
Callout Label	Unconnected Component
Tool-tip: Level 1 Header	Validation Error
Tool-tip: Level 1 Text	Unconnected Component
Tool-tip: Level 2 Text	All components must be connected to a schematic line. For example, inline elements such as valves must be placed somewhere on a line run, while end-line elements such as tanks and vessels should be connected to the end of a line.
Tool-tip: Footer	Press F1 for more Help

Unresolved Off-page Connectors

Unresolved off-page connector errors occur when users place an off-page connector, but fail to mate it to a connector in another drawing. When validation is configured to check for these errors, a markup is placed around each offending connector symbol.

The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

Element	Text
Callout Label	Unresolved off-page connector
Tool-tip: Level 1 Header	Validation Error
Tool-tip: Level 1 Text	No connection to a symbol in another drawing
Tool-tip: Level 2 Text	Off-page connectors must be mated to another connector appearing in another drawing.
Tool-tip: Footer	Press F1 for more Help

Base AutoCAD Objects

Base AutoCAD objects, particularly in the case of non-P&ID blocks, may be misinterpreted by users as P&ID objects when they appear in drawings. The danger here is that, as non-P&ID object types, they will not be captured in any reports. By default, Cork is configured to check for the presence of non-P&ID blocks, but users are free to configure other native AutoCAD objects as validation errors as well.

Note that while some P&ID object types are blocks, it is possible to differentiate our objects from non-P&ID blocks. If possible, it would also be desirable to exclude any title-block blocks from the validation check as well, but this is of lesser importance than excluding P&ID blocks.

When a base AutoCAD object error is returned, a callout bubble is drawn around the offending object.

The following table describes the elements that comprise the callout label and its associated enhanced tool-tip.

Element	Text
Callout Label	Base AutoCAD object
Tool-tip: Level 1 Header	Validation Error
Tool-tip: Level 1 Text	Base AutoCAD object: <*object_type*>
Tool-tip: Level 2 Text	Base AutoCAD objects, particularly blocks, can be problematic in that they can resemble native P&ID objects such as valves or tanks, but as non P&ID objects they are not captured in any generated reports. By default, Validation checks for the presence of AutoCAD blocks, but you can also flag the presence of

Element	Text
	other base AutoCAD objects as error conditions if desired.
Tool-tip: Footer	Press F1 for more Help

4.1.1.1.4 Error Zoom Behavior

As users select individual errors from Project Manager, AutoCAD makes the parent DWG active and zooms to the error object in the drawing editor. The zoom operation is based on the extents of the callout bubble of the erroneous object, with a padding factor of 50%. This is equivalent to the following AutoCAD operations:

- Zoom Object
- Zoom Scale .5 x

Different errors select and zoom to different objects in the drawing editor, as detailed in the following table.

Error	Objects selected
AutoCAD primitive	AutoCAD object
Property mismatch	Erroneous in-line or end-line component
Non-terminating line	Non terminating segment. If the line doesn't terminate on both ends, this is reported as two separate validation errors (one error per end segment).
Un-annotated line	Center point of the un-annotated line
Incomplete tag	Parent object
Unconnected component	Component
Flow conflicts (line connecting junction)	All conflicting line segments intersecting at the conflicting junction
Flow conflicts (in-line component)	In-line component
Non mated off-page connector	Off-page connector

4.1.2 User Objects

4.1.2.1 Feature Object Model

The Validation feature does not introduce any new objects, other than the validation markup pseudo-object that is described earlier in this document.

4.1.2.1.1 Object, Properties and Actions Table

4.1.3 User Actions

The following list catalogs the various actions users need to accomplish using this feature. The table serves as a convenient reference catalog for the range of functionality required by this feature. Please note that detailed explanations of the various actions are captured in later sections of this document, beginning in section 4.2.

Root Action	Additional Actions
Plotting or publishing	• Configure whether these actions triggers an automatic validation when initiated from within a project context • If errors exist, provide users with opportunity to review them prior to initiating the root action via the Project Manager or the Dashboard o Ability to toggle the display of individual errors so they don't show up in plotted output or in the drawing editor • Markup capabilities to visually indicate problem states in drawings

Configuration	• Invoke the Drawing Validation Settings dialog box to configure the following settings:
	  o Whether validation operations are triggered when plotting or publishing
	  o Which potential sources of validation errors are checked (Base AutoCAD objects, size mismatches, spec mismatches, non-terminating lines, connections, flow direction, tags)
	  o Whether background validation is enabled
	  o Polling frequency for background validation

User-initiated Validation	• Validate one or more project drawings – if errors exist, present them in the Project Manager for review o Toggle display of individual errors
Background validation	• Configure bubble notification settings • Dismiss bubble notification • Refresh validation markups in open project drawings
Saving	• If any new validation errors introduced during the current editing session, refresh validation markups to indicate new error states

4.2 Interaction Design Inventory

4.2.1 Command Checklist

Command Name	Invokes UI Only	New/Modified/ Removed	New Commands	
			CUI Name and Description	PGP Alias
VALIDATE	Yes	New	N/A	N/A
VALIDATECONFIG	Yes	New	N/A	
PLOT	TBD	Modified	N/A	N/A
PUBLISH	TBD	Modified	N/A	N/A

4.2.1.1 VALIDATE

Item	Description/Value
Name	VALIDATE
Description	Performs a validation operation of the current project.
Allowable Context	Can only be invoked when a P&ID project is loaded
Accessed from (CLI, UI Element, etc.)	CLI, File > Validate Project menu option, P&ID toolbar Validate Project menu options, context menu exposed through Project Manager

The VALIDATE command is a purely UI-driven command: it is exposes no command line prompts and accepts no arguments. When the command is invoked with a project currently loaded into the Project Manager, the validation operation commences, and the Validation Progress dialog box is displayed. If the command is invoked in the absence of a current project, the following error prompt is displayed on the command line:

```
Command: Syntax error – use PROJECTMANAGER to load a project, then run VALIDATE
again.
```

4.2.1.2 VALIDATECONFIG

Item	Description/Value
Name	VALIDATECONFIG
Description	Invokes the Validation Settings dialog box.
Allowable Context	No restrictions
Accessed from (CLI, UI Element, etc.)	CLI, File > Configure Validation Settings menu option, Validation panel of Dashboard, context menu option exposed through Project Manager

VALIDATECONFIG is a purely UI-driven command – it exposes no command prompts and accepts no arguments. When invoked, it launches the Validation Settings dialog box.

4.2.1.3 PLOT

Item	Description/Value
Name	PLOT
Description	Legacy AutoCAD command. Cork will expose configuration options that initiate a validation operation when plot or publish operations are invoked.
Allowable Context	No restrictions
Accessed from (CLI, UI Element, etc.)	No changes versus standard AutoCAD access mechanisms

PLOT is a legacy AutoCAD command. When it is invoked, the P&ID application must determine if the drawing being plotted is part of the current project. If so and validation is configured to perform on plot/publish operations, a validation operation must be performed on the drawing prior to plotting it. If any validation errors are encountered, they must be recorded to the plot/publish log file as detailed elsewhere in this specification.

4.2.1.4 PUBLISH

Item	Description/Value
Name	PUBLISH
Description	Legacy AutoCAD command. Cork will expose configuration options that initiate a validation operation when plot or publish operations are invoked.
Allowable Context	No restrictions
Accessed from (CLI, UI Element, etc.)	No changes versus standard AutoCAD access mechanisms and those exposed through Project Manager for Harp

PUBLISH is a legacy AutoCAD command. Similar to PLOT, the P&ID application must determine if any drawings in the publish set belong to the current project when PUBLISH is invoked. If so and configured to check for this case, a validation operation is performed on any drawings that are members of the current project. Any drawings containing validation errors must log this to the plot/publish log file as detailed elsewhere in this specification.

4.2.2 System Variables Checklist

SYSVAR Name	New/Modified/Removed
PIDBGVALIDATE	New
PIDBGVALIDATETIME	New
PIDMARKUPARC	New

4.2.2.1 PIDBGVALIDATE

Item	Description
Name	PIDBGVALIDATE
Description	Controls whether background validation of project drawings is enabled. Note that this is a global, per user setting that controls whether background validation is enabled for all projects. No mechanism is exposed for setting this variable on a per-project basis.
Data Type	Integer
Values	0 – Background validation is turned off 1 – Background validation is turned on. Changing the value of this system variable causes the analogous control exposed through the P&ID Validation Settings dialog box to toggle as well.
Initial Value	1
Stored in	Registry
Accessed from (CLI, SETVAR, UI location, etc.)	CLI, SETVAR command, P&ID Validation Settings dialog box

4.2.2.2 PIDBGVALIDATETIME

Item	Description
Name	PIDBGVALIDATETIME
Description	Sets the polling frequency for automatically validating project drawings.
Data Type	Real
Values	Any real number between 10 and 1440. Specifies the time (in minutes) between automatically triggered validation operations.
Initial Value	15
Stored in	Registry
Accessed from (CLI, SETVAR, UI location, etc.)	CLI, SETVAR command, P&ID Validation Settings dialog box

4.2.2.3 PIDMARKUPARC

Item	Description
Name	PIDBMARKUPARC
Description	Sets the length of arcs in the validation markup.

Data Type	Real
Values	Any real number between .001 and 5. Specifies the size (in current units) of the length of each arc that constitutes the markup bubble.
Initial Value	Varies: .1 for imperial drawings, 2.5 for metric
Stored in	Drawing
Accessed from (CLI, SETVAR, UI location, etc.)	CLI, SETVAR command

The PIDMARKUPARC specifies the length of each arc that comprises a markup bubble. Each template that ships with Cork must be updated to include the appropriate value (metric vs. imperial) for this system variable.

4.2.3 User Interface Checklist

Below is a listing of all dialogs, ESWs, and other user interface elements. Task dialogs are handled in a separate section:

UI Element Name	Brief Description
P&ID Validation Settings Dialog Box	Provides an environment for users to specify settings related to validation operations.
Validation Progress Dialog Box	Progress meter that is displayed while a foreground[12] validation operation is being processed.
Project Manager ESW	Legacy P&ID dialog. A new Validation Errors panel is added to the bottom section of the window. This panel lists any validation errors for the current node selection from the PM tree.
Plot and Publish Details Dialog Box	Legacy AutoCAD dialog box. It is updated to become contextually aware of Cork validation errors.
Dashboard	A new Validation panel is exposed for each of the out-of-box Process and Power workspaces (P&ID PIP, P&ID ISO, P&ID ISA).

4.2.3.1 P&ID Validation Settings

The P&ID Validation Settings dialog box provides an environment for users to configure which potential problem sources should be reported when the drawing is validated. These settings are global, and apply to all P&ID projects. The dialog box is modal and resizable, but can't be sized smaller than its initial size.

[12] Background validation operations do not display a progress meter.

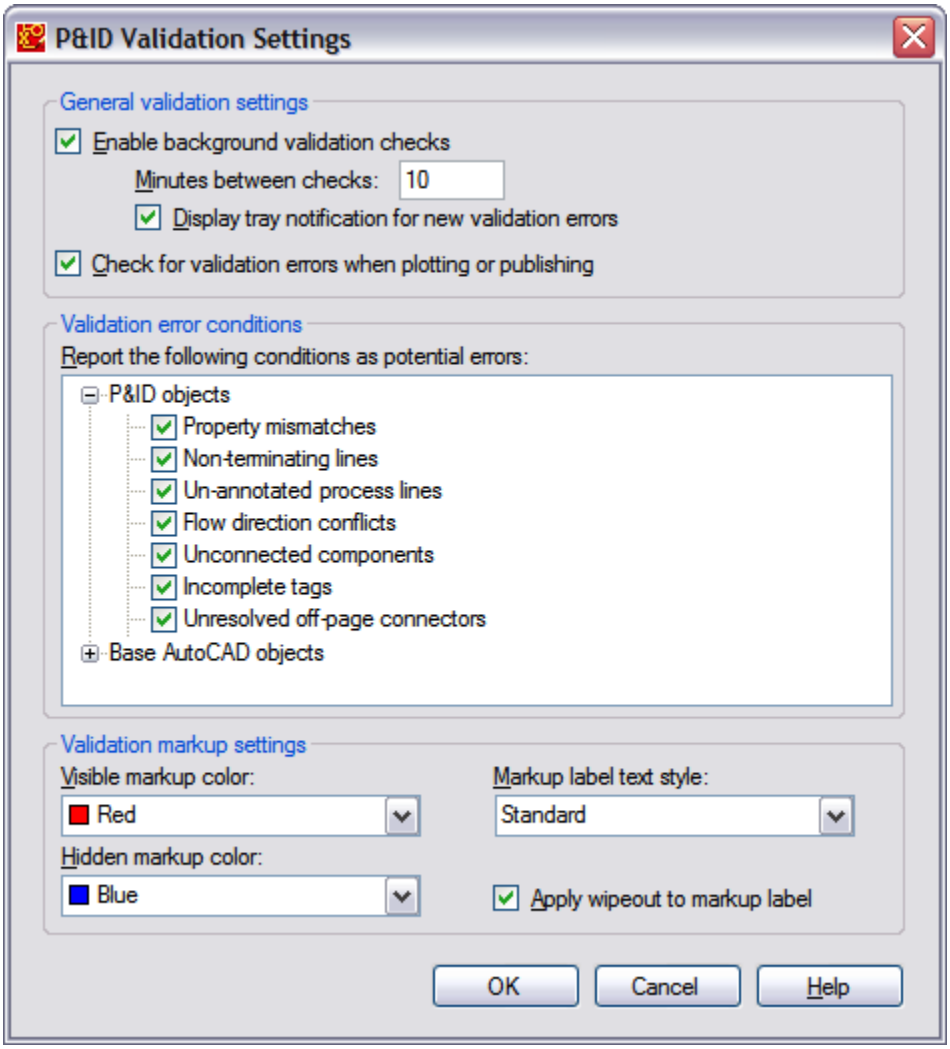
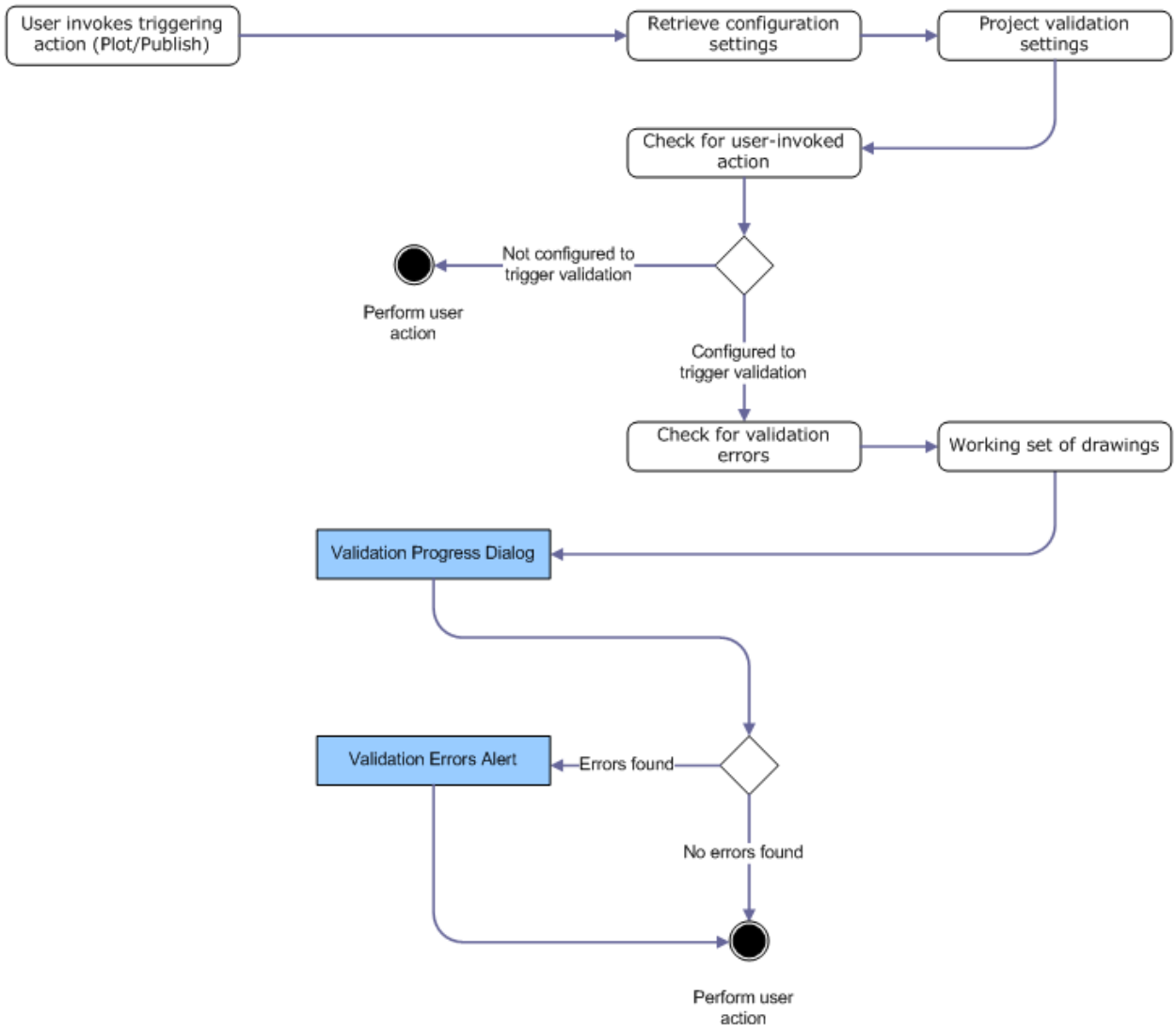


Figure 4: P&ID Validation Settings Dialog Box

The settings exposed through this dialog box control the behavior of Cork in response to certain actions, as depicted in the following screenshots.

AD 02: Implicit Validation



By default, Cork checks for all P&ID errors and the presence of non-P&ID blocks[13]. Users can toggle the checks for these AutoCAD objects individually, or can also choose an "All AutoCAD Objects" option, which will consider the presence of any base AutoCAD object an error.

[13] While most P&ID objects are exposed as their own object types, there are a number that are exposed as AutoCAD blocks. For the purpose of validation errors, only non-P&ID blocks should be reported as validation errors. More specifically, any P&ID object that is reported as a block through the Properties Palette, LIST command, etc., is not treated as a validation error.

The following table will be completed during the specification phase, detailing all controls exposed through this dialog box. All controls are presented in their actual tab sequence. Accelerators (where applicable) are indicated via underlined text.

ID	Control Type	Text	Enabled/Disabled	Default Value	Description
	Check box	Enable background validation checks	Enabled	Checked	Toggles whether validation checks are performed automatically in the background. Changing this option toggles the value of the PIDBGVALIDATE system variable.
	Edit box	Minutes between checks	Enabled	10	Sets a timer that counts down the number of minutes until the next background validation takes place (if background validation is enabled). Users can enter any integer value between 1 and 1440 (the number of minutes in a day). If users enter a value outside of this range, the value resets itself to the last entered value. When users mouse over this control, the following tool-tip is displayed: “Enter an integer value between 1 and 1440”. If users clear the “Enable background validation checks” check box, this control is disabled.
	Check box	Display tray notification for new validation errors	Enabled	Checked	Toggles whether a tray notification is displayed in the AutoCAD system tray when new validation errors are found in any current project drawings that are open for editing by the current user. Note that the tray notification is only triggered for new validation errors: any previously encountered errors will not trigger a notification. Also note that the notification is only displayed for drawings that are actively opened for editing by the current user.
	Check box	Check for validation errors when plotting or publishing	Enabled	Checked	Toggles whether a foreground validation operation is initiated when users plot or publish project drawings.
	Tree control (single select)	Report the following conditions as potential errors:	Enabled	N/A	Provides an environment for users to indicate which potential error conditions are checked. These are global settings for the current user profile. . If needed, a vertical scrollbar is added to the right side of the tree. The following nodes are exposed: • P&ID objects ○ Property mismatches

ID	Control Type	Text	Enabled/Disabled	Default Value	Description
					○ Non-terminating lines ○ Un-annotated process lines ○ Flow direction conflicts ○ Unconnected components ○ Incomplete tags ○ Unresolved off-page connectors • Base AutoCAD objects ○ All AutoCAD objects ○ Blocks ○ Lines ○ Polylines ○ Circles ○ Annotations When initially invoked, the P&ID Objects node is expanded and the Base AutoCAD Objects node collapsed. The Property Mismatches child node is selected in the tree. Each sub node is preceded by a check box (top level parents do not have check boxes). A single node is always selected in the tree control. A brief helper string related to the current selection is displayed as a tool-tip when users hover over a given error type. By default, all P&ID Objects check boxes are checked, and in Base AutoCAD Objects the Blocks check box is checked. When the All AutoCAD Objects check box is checked, the control becomes disabled and the check marks of the other nodes in the Base AutoCAD Objects node are checked. As soon as the user clears one of the other check boxes, the All AutoCAD Objects check box is also cleared and once again becomes enabled. Right-clicking a top-level node in the tree reveals a shortcut menu with the following options: • **Check All** – Checks all child node check marks. If all nodes are currently checked, this option is disabled. • **Clear All** – Clears all child node check marks. If all nodes are currently cleared, this option is disabled. The following tool-tips are displayed when users mouse over a given node in the tree:

ID	Control Type	Text	Enabled/Disabled	Default Value	Description
					• **P&ID objects** – "Errors specific to P&ID objects and functionality, such as size mismatches between components and schematic lines." • **Property mismatches** – "Property mismatches between schematic lines and associated components." • **Non-terminating lines** – "Schematic lines that don't connect to components or end with a termination symbol." • **Un-annotated process lines** – "Lines from the Pipe Lines Tool Palette grouping that have not been annotated in the P&ID." • **Flow direction conflicts** – "Conflicts in the flow of a material through a process line." • **Unconnected components** – "Inline or end-line components that are not connected to a schematic line." • **Incomplete tags** – "Any tag element with missing data." • **Unresolved off-page connectors** – "Off-page connectors that don't continue on another project drawing file." • **Base AutoCAD objects** – "Errors related to the presence of base AutoCAD objects, which can be misinterpreted as P&ID objects in plotted output." • **All AutoCAD objects** – "Flags the presence of any AutoCAD object as an error." • **Blocks** – "Non-P&ID AutoCAD blocks (xrefs, blocks, and dynamic blocks – can potentially be misinterpreted as P&ID components)." • **Lines** – "AutoCAD line (single and multi) objects (can potentially be misinterpreted as P&ID lines)." • **Circles** – "AutoCAD circle objects (can potentially be misinterpreted as instrument bubbles)." • **Annotations** "AutoCAD text, mtext, dtext, leaders, dimensions, attributed, tables, hyperlinks (can potentially be misinterpreted as P&ID annotations)."
	Drop-down list	Visible markup	Enabled	Red	Specifies the color that is applied to visible markups. The following values are initially available:

ID	Control Type	Text	Enabled/Disabled	Default Value	Description
		color:			• Red • Yellow • Green • Cyan • Blue • Magenta • White • Select Color… The "White" option toggles between White and Black depending on the 2D background color setting exposed through the Options dialog box for the current space (model or paper). If the background color is set to Black, the color exposed through the list is White, and if set to White, the color exposed is Black. The Select Color option brings up the standard AutoCAD Color dialog box (for example, the same dialog box that is invoked by selecting the Select Color option from the Color Control drop-down list available from the Properties toolbar exposed at the top of the application window.
	Drop-down list	Hidden markup color:	Enabled	Blue	Specifies the color that is applied to hidden markups. All values exposed through the Visible Markup Color control described above are also available here, and behave in an identical manner.
	Drop-down list	Markup label text style:	Enabled	Standard	Specifies the text style that is applied to the markup text. Any style exposed through the standard AutoCAD Style dialog box is available. UA should document best-practices with respect to setting the text height through this dialog box (use a value of .10 for imperial drawings, and a value of 0 for metric).
	Check box	Apply wipeout to markup label	Enabled	Checked	Toggles whether a wipeout is applied to objects appearing below the markup text and border.

4.2.3.2 Validation Progress Dialog Box

The Validation Progress dialog box is presented when users initiate a foreground validation operation, either implicitly because of configuration settings (e.g., when plotting or publishing) or explicitly through a user-initiated validation. Note that the dialog does not display for background process validation polling. The dialog box surfaces two progress meters

letting users know the overall progress in validating the working set of drawings, along with a Cancel button to terminate the validation. For implicit operations, clicking Cancel also cancels the calling operation (e.g., plotting or reporting). While this may be regarded as a nuisance to users, if the project is configured to validate drawings when performing these operations, users should not be able to circumvent this validation without reconfiguring the project settings.

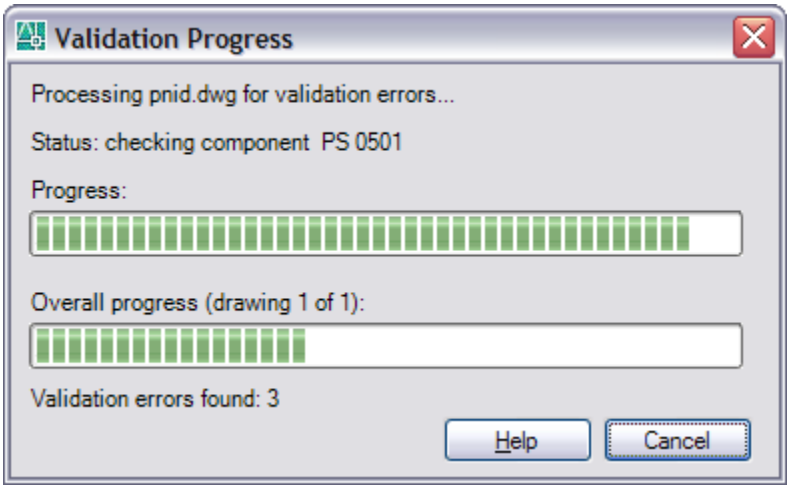


Figure 5: Validation Progress Dialog Box

The dialog box exposes a number of dynamic labels. These labels and the other dialog controls are described in the table below.

Control	Description
Dynamic label	Indicates the drawing currently being processed. The label takes the following format: "Processing <*DWG_name*>.dwg for validation errors…"
Dynamic label	Indicates the current status of the validation check. Depending on the current validation settings, the current drawing will be checked for a variety of P&ID validation errors, as well as the presence of AutoCAD primitives and blocks. P&ID errors are checked first (if configured to be checked), followed by AutoCAD primitives (if checked). When checking for P&ID validation errors, an individual label taking the following format is displayed for each P&ID object matching the current configuration settings: "Status: checking P&ID <*object_type*> <*component_tag*>" When checking for AutoCAD primitives, a label taking the following format is displayed: "Status: checking for disallowed AutoCAD primitives (<*x*> found)", where *x* is a number initially set to 0 which increments with each disallowed primitive the audit identifies.
Progress meter	Indicates the progress in searching the current drawing.
Progress meter	Indicates the progress in searching the complete set of drawing files.
Help	Launches the application Help file, displaying a topic related to the P&ID validation process.

4.2.3.3 Project Manager ESW

The Project Manager interface provides centralized and convenient access to the drawings that comprise a P&ID project. In Harp, the Project Manager is updated to include a Validation Errors panel. This panel displays all validation errors associated with the current node selection in the Project Manager tree. If the project node is selected, all errors across the entire project are displayed.

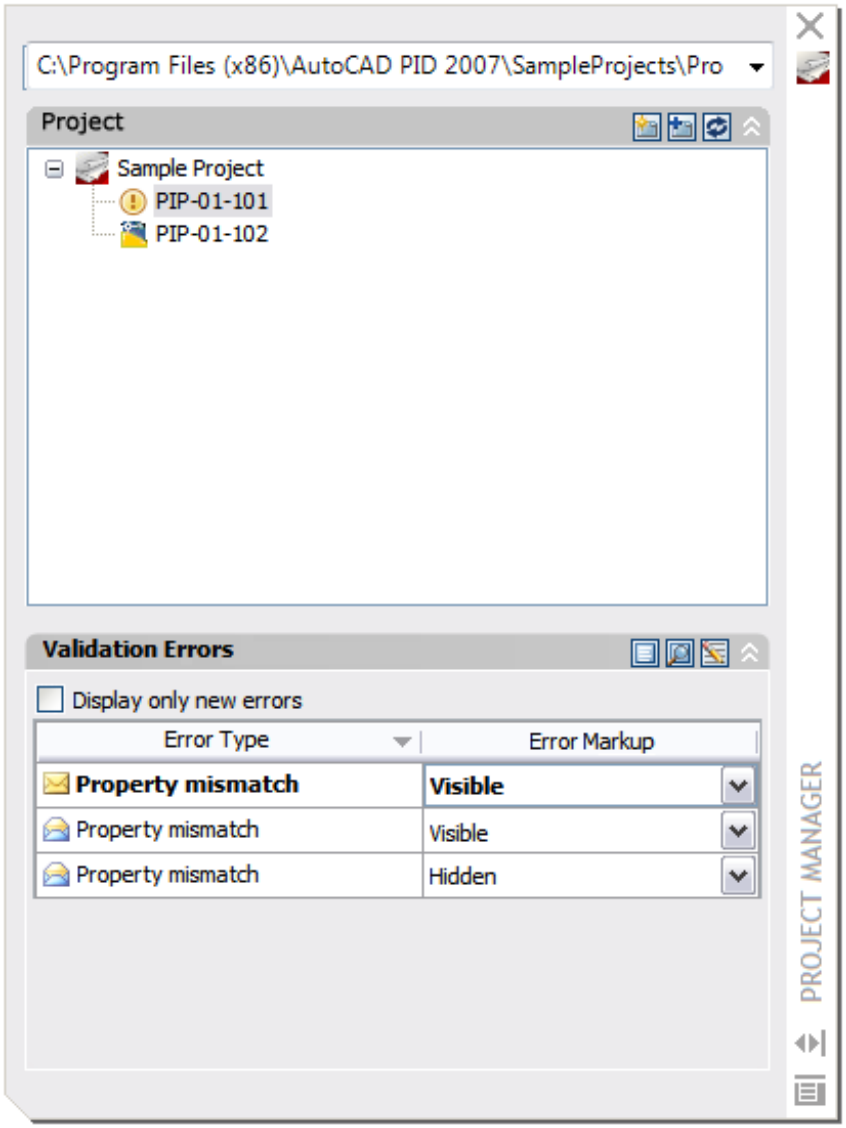


Figure 6: Validation Errors Panel of Project Manager

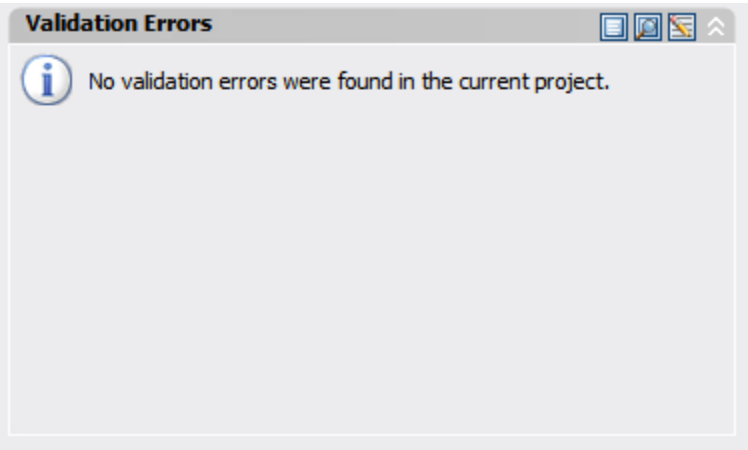
Each error in the error grid is preceded by an icon, indicating whether the error is new and unviewed or has been previously been viewed. New errors are added to the top of the display for a given node, pushing previous errors down in the list. As soon as users click on a validation error from the panel to view it, the icon toggles to the previously viewed icon.

As users select individual errors from the Validation Errors panel, the associated object is selected and zoomed to in the drawing editor[14]. Only a single item can be selected at a time in the Validation Errors grid. Users can sort on either the Error Type or Error Visibility columns by clicking the column header. Clicking a second time on a given column header performs a reverse sort.

The Error Visibility column allows users to determine whether each individual error has a markup associated with it and always visible in the drawing editor environment. If an error is marked as Visible, a markup balloon and callout text will be present in the drawing editor environment, and visible when user plot out the associated drawing file. If an error is marked as Hidden, the markup balloon and callout text are suppressed in the drawing environment and not visible when the drawing is plotted. When users select a Hidden validation error from the panel, however, a modified rendition of the markup balloon and callout text are displayed until a new error is selected in the panel. The visual appearance of the markup will be altered in some way (TBD during specification phase) to clearly distinguish them from Visible markups[15]

This new panel becomes the default panel when the Project Manager is initially invoked. Each time users invoke the Project Manager in a Cork session, the Validation Errors panel is restored, regardless of the previous panel last active in the previous editing session.

If a given project has no validation errors currently associated with it, the grid control is dynamically replaced by an icon and text string indicating that the project contains no errors, as depicted in the following screenshot:



Similarly, if one or more drawing nodes are selected in the Project Manager tree and the selection set contains no errors, the string would update as appropriate (“No validation errors were found in the selected drawing” or “No validation errors were found in the selected drawings”).

The following table describes the new controls that are proposed for the Validation panel. Additional detail about the various controls will be provided during the specification phase.

[14] If the drawing containing the object isn’t already opened, Cork opens the drawing.

[15] For example, the markups and associated callouts might be presented in a different color, or with some sort of opacity or screening applied to them.

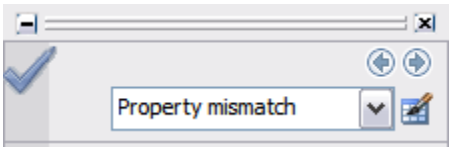
ID	Control Type	Text	Enabled/Disabled	Default Value	Description
1	Button	Tool-tip: "Validation Errors"	Enabled	N/A	Note that this button has not yet been added to the Project Manager screenshot depicted in this section. The button will be developed by our graphics team during the specification phase. Toggles the Details region of the Project Manager to display the Validation Errors panel. This new panel will become the default panel for the Cork release.
2	Label	Varies	N/A	N/A	Dynamic text. If the project node is selected and there are no errors in the current project, the following string is displayed: "No validation errors were found in the current project." If a single drawing with no errors is selected, the following string is displayed: "No validation errors were found in the selected drawing." If multiple drawings with no errors are selected, the following string is displayed: "No validation errors were found in the selected drawings." If the current selection does contain errors, the label region of the panel is replaced by a check box and a grid control, as described in the following rows.
3	Check box	Display only new errors	Enabled	Unchecked	Toggles whether the error list displays only new errors or all errors. When checked, any previously viewed error is removed from the list.
4	Grid	N/A	Enabled	N/A	Two column grid. When errors are present in the selected node(s), each error is presented as a separate row. Clicking on an individual row opens the parent drawing (if not already open) and zooms to the specific error condition. If the Error Markup column for the selected error is set to visible, the markup is displayed using the Visible Markup Color (by default red) specified in the P&ID Validation Settings dialog box. If the Error Markup column for the selected error is set to hidden, the markup is suppressed, but temporarily toggled on when users select the error row from the Project Manager. For the duration of the time the row is selected, the markup is displayed using the Hidden markup color specified in the P&ID Validation Settings dialog box. As soon as the user moves to a new error record, the markup toggles back to hidden and is no longer displayed in the drawing editor. The Error Type column displays a non-editable string indicating the type of error. When users mouse over this column, the following tool-tip is displayed: "Indicates the condition that triggered the validation error". Each error

ID	Control Type	Text	Enabled/Disabled	Default Value	Description
					is preceded by an icon – when an error is new and has not been previously viewed, the following icon is used and text in the error row is in bold face:  . If the error has previously been viewed, the following icon is used and text in the error row is in plain face:  . Through the Error Markup column, users can hide individual errors so they don't show up in plotted output or general viewing when interacting with the drawing in the drawing editor. When users mouse over this column, the following tool-tip is displayed: "Toggles the visibility state of the markup bubble and label that is associated with the error object." Users can apply a sort based on either column by clicking the column header. A reverse sort is applied by clicking the header a second time.

When users right-click the Project node from the Project Manager, a new Validate… (accelerator = V) menu option is exposed, which invokes a validation operation. This menu option appears before the existing Audit Project option.

4.2.3.4 Dashboard

In Cork, it is expected that various features will leverage the Dashboard infrastructure by exposing panels appropriate for interacting with P&ID drawings[16]. The Validation feature will expose a small panel, as depicted in the following screenshot.



The Validation panel will list all validation errors in the active project drawing. If a project drawing is not active in the drawing editor, or if the active drawing contains no validation errors, all controls except the P&ID Validation Settings button are disabled. If a project drawing is active with one or more validation errors, these errors are presented in a drop-down list. Users can also step through the set of errors sequentially via a pair of navigation buttons exposed through the interface.

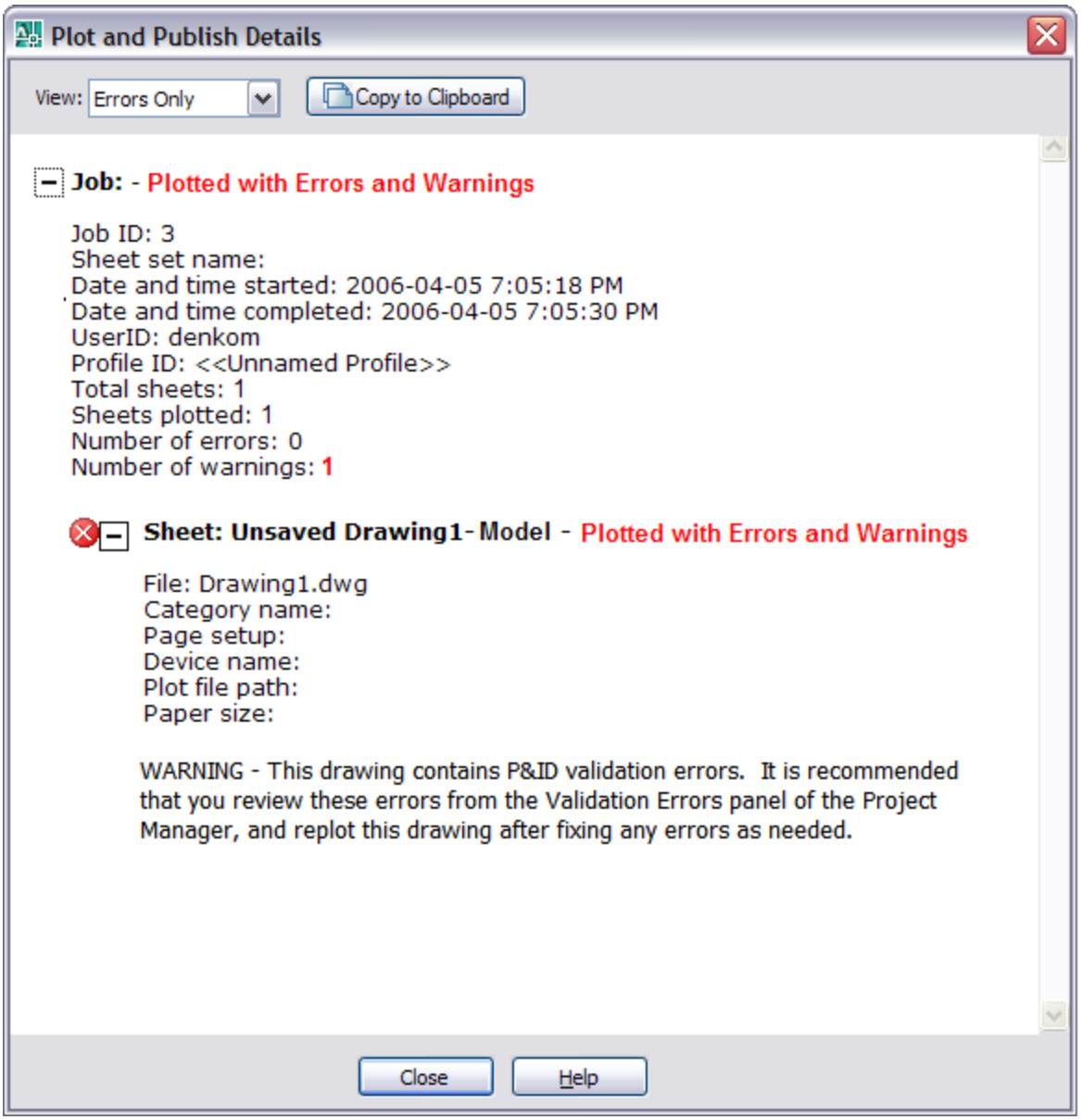
The following table describes the controls that are proposed for the Validation panel. Additional detail about the various controls will be provided during the specification phase.

[16] The final scope of Cork is still unknown at the time this CD was authored, and the original scope has been significantly curtailed over the writing of this document. A determination as to whether there is a sufficient feature set to warrant exposing P&ID Dashboard content will be made during the detailed specification phase.

ID	Control Type	Text	Enabled/Disabled	Default Value	Description
1	Button	Tool-tip: "P&ID Validation Settings"	Enabled	N/A	Invokes the P&ID Validation Settings dialog box.
2	Button	Tool-tip: "Previous Validation Error"	Disabled	N/A	Navigates to the previous validation error for the current drawing. This button is initially disabled until the user moves to a different navigation error. When enabled, this button sequentially steps back through each error in the list of errors from the current error. If there are no errors in the current drawing, this control is disabled.
3	Button	Tool-tip: "Next Validation Error"	Enabled	N/A	Navigates to the next validation error for the current drawing. When there are no errors in the current drawing or the last error in the list is reached, this button is disabled.
4	Drop-down list	N/A	Enabled	N/A	Lists the validation error in the active project drawing. If no validation errors are present or a non-project drawing is active, the list is empty and the control is disabled. If users use the navigation buttons to traverse the set of validation errors, the list updates with each button press to display the current validation error.
5	Button	Tool-tip: TBD during spec phase	Enabled	Toggled off	Toggle button – sets the visibility for the current error in the error list.

4.2.3.5 Plot and Publish Details Dialog Box

The Plot and Publish Details dialog box was introduced as part of the Background Plot feature in the AutoCAD Neo release. It is updated to become contextually aware of Cork validation errors. If users have configured their projects to not pre-validate project drawings prior to plotting or publishing them, they should still be checked on an individual basis as they are being output, with any drawings that contain validation errors logging this as a warning for the source DWG in the Plot and Publish log. Similarly, if users have configured a project to pre-validate but they choose to explicitly ignore a pre-plot operation warning, any drawings that are problematic must be captured in the log.



The following primary changes are required for the Plot and Publish Details log:

- Any drawings that are plotted or published from the current project must be checked for validation errors when they are output
- Any drawings that contain validation errors must add a warning entry into the plot log
- The following new warning string is exposed: “WARNING – This drawing contains P&ID validation errors. It is recommended you review these errors from the Validation Errors panel of the Project Manager, and replot this drawing after fixing any errors as needed.”

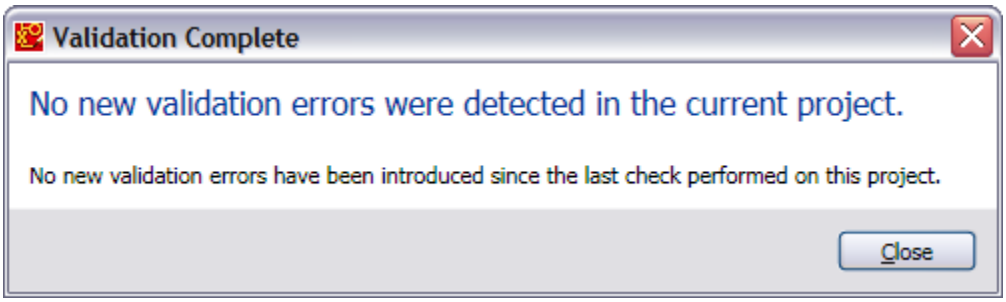
4.2.4 Task Dialogs and Messages

Inventory of Task Dialogs:

Name	Brief Description
Validation Complete	Displayed when users explicitly invoke a validation operation that returns no errors.
4.2.4.1 New Validation Errors Detected	Displayed when errors are detected through a user-initiated validation operation.
Plot Job Validation Errors	Displayed if plot/publish validation checks are configured and errors were detected during the subsequent validation operation.

4.2.4.2 Validation Complete

The Validation Complete alert is displayed if a user-initiated validation operation returns no new errors.

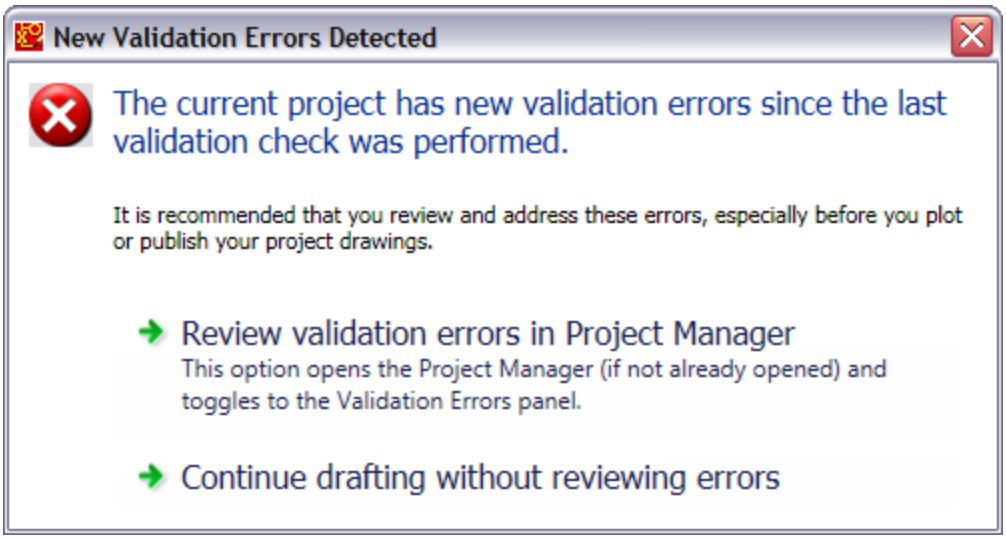


Item	Visible	Text/Explanation
Title	Always Visible	Validation Complete
Main Instruction	Yes	No validation errors were detected in the current project.
Image Type (Warning, Error, Information, Question, Custom)	TBD (Information icon desirable)	
Content	Yes	No new validation errors have been introduced since the last check performed on this project.
Expanded Content	No	
Custom Buttons – Enumerate any custom buttons below.		
	Visible	Action When Clicked
Custom Button 1,	No	

Item	Visible	Text/Explanation	
e.g. “Continue”			
Standard Buttons – If you are using Command Links the only Standard Button you can use is Cancel.			
	Visible	Action When Clicked	
OK	No		
Yes	No		
No	No		
Retry	No		
Cancel	No		
Close	Yes	Dismisses the task dialog	
Other Items			
	Visible	Text and Explanation	
“Do not show me this again” Checkbox	No		
Footer			
	Visible	Icon Type	Text
“Footer” Text	No	Error, Warning or Information	Specify Text and Hyperlink

4.2.4.3 New Validation Errors Detected

The New Validation Errors Detected alert is displayed when users invoke an explicit Validation operation that returns new errors that are not already indicated in the Project Manager.

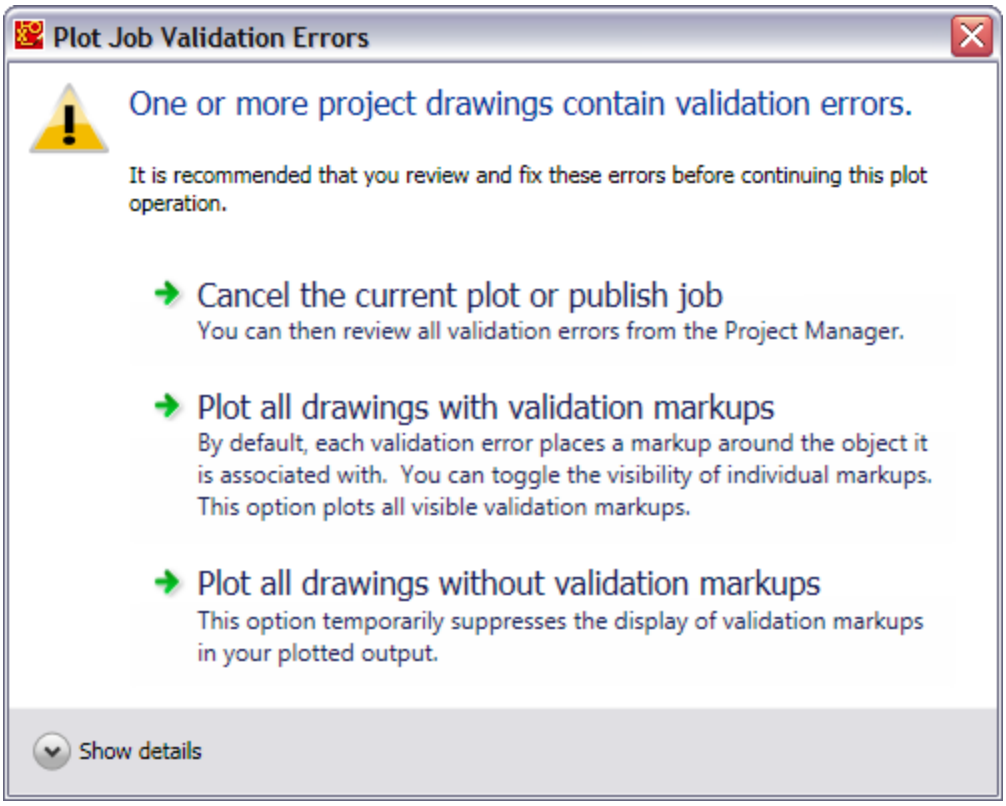


Item	Visible	Text/Explanation
Title	Always Visible	New Validation Errors Detected
Main Instruction	Yes	The current project has new validation errors since the last validation check was performed.
Image Type (Warning, Error, Information, Question, Custom)	Error	
Content	Yes	It is recommended that you review and address these errors, especially before you plot or publish your project drawings.
Expanded Content	No	
Command Link #1	Dismisses the alert and performs the following additional operations: • If the Project Manager is not already open in the drawing editor, open it • If the Validation Errors panel is not the active panel, toggle it active	
Text	Review validation errors in Project Manager	
Explanation	This option opens the Project Manager (if not already opened) and toggles to the Validation Errors panel.	
Command Link # 2	Dismisses the alert without performing any additional actions. Has no effect if the Project Manager is already open in the drawing editor.	
Text	Continue drafting without reviewing errors	
Explanation		
Custom Buttons – Enumerate any custom buttons below.		
	Visible	Action When Clicked

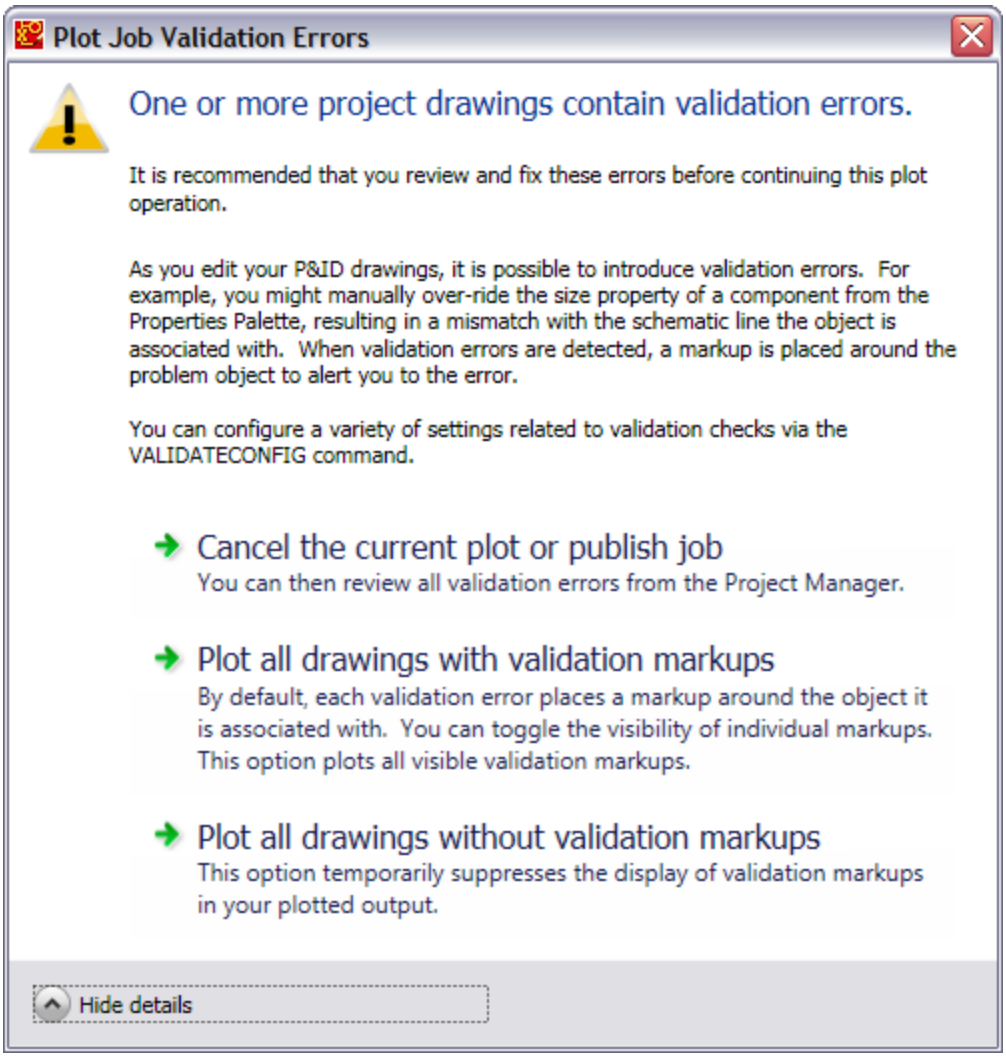
Item	Visible	Text/Explanation	
Custom Button 1, e.g. “Continue”	No		
Standard Buttons – If you are using Command Links the only Standard Button you can use is Cancel.			
	Visible	Action When Clicked	
OK	No		
Yes	No		
No	No		
Retry	No		
Cancel	No		
Close	No		
Other Items			
	Visible	Text and Explanation	
“Do not show me this again” Checkbox	No		
Footer			
	Visible	Icon Type	Text
“Footer” Text	No	Error, Warning or Information	Specify Text and Hyperlink

4.2.4.4 Plot Job Validation Errors

The Validation Errors alert is displayed if validation checks are configured for plot/publish operations and errors are detected while performing a plot/publish operation.



If users click the Show Details button, additional information is displayed in the task dialog as depicted below:



Item	Visible	Text/Explanation
Title	Always Visible	Validation Errors
Main Instruction	Yes	One or more project drawings contain validation errors.
Image Type (Warning, Error, Information, Question, Custom)	Warning	
Content	Yes	It is recommended that you review and fix these errors before continuing this plot operation.
Expanded Content	Yes	As you edit your P&ID drawings, it is possible to introduce validation errors. For example, you might manually over-ride the size property of a component from the Properties Palette, resulting in a mismatch with the schematic line the object is associated with. When validation errors are detected, a markup is

Item	Visible	Text/Explanation
		placed around the problem object to alert you to the error. You can configure a variety of settings related to validation checks via the VALIDATECONFIG command.
Command Link #1	When this link is clicked, the message box is dismissed and the current plot/publish operation is canceled.	
Text	Cancel the current plot or publish job	
Explanation	You can review all validation errors from the Project Manager.	
Command Link #2	Continue the current plot/publish job. Any markups in project drawings with their visibility state enabled will be plotted as part of the operation and visible in the plotted output.	
Text	Plot all drawings with validation markups	
Explanation	By default, each validation error places a markup around the object it is associated with. You can toggle the visibility of individual markups. This option plots all visible validation markups.	
Command Link #3	Continue the current plot/publish job. All markups (regardless of their current visibility state) are suppressed in the plotted output. Note that this is a temporary suppression for the duration of the current plot/publish job only – all markups maintain their visibility settings within the drawing editor and on subsequent plot/publish operations.	
Text	Plot all drawings without validation markups	
Explanation	This option temporarily suppresses the display of validation markups in your plotted output.	
Custom Buttons – Enumerate any custom buttons below.		
	Visible	Action When Clicked
Custom Button 1, e.g. “Continue”	No	**N/A**
Standard Buttons – If you are using Command Links the only Standard Button you can use is Cancel.		
	Visible	Action When Clicked
OK	No	N/A
Yes	No	N/A
No	No	N/A
Retry	No	N/A
Cancel	No	N/A
Close	No	N/A

Item	Visible	Text/Explanation	
Other Items			
	Visible	Text and Explanation	
"Do not show me this again" Checkbox	Yes or No	N/A	
Footer			
	Visible	Icon Type	Text
"Footer" Text	Yes or No	Information	Specify Text and Hyperlink

4.3 Access

4.3.1 Menus, Toolbars, and Dashboard Control Panels

4.3.1.1 Application Menu

The Validation feature exposes two new options from the File menu for each of the P&ID workspaces (P&ID PIP, P&ID ISO, & P&ID ISA):

- Validate Project
- Configure Validation Settings…

These menu options are described in greater detail in the table below. Accelerators (where applicable) are indicated in the Text column as underlined text.

Text	Enabled/Disabled	Description
File > Validate Project[17]	Varies	Invokes the VALIDATE command, which performs a validation operation on the current project. If no project is currently loaded in the Project Manager, this menu option is disabled. This menu option appears immediately following the existing Open Project menu option.

[17] Note that this is a duplicate accelerator to the existing Plot Preview menu option. The Microsoft UI standards specify that it is acceptable to duplicate accelerators, and when this occurs, cycling between the two menu options takes place. The Validate Project menu option, appearing first in the menu and of greater importance in the context of P&ID workspaces, will be the first option cycled to.

Text	Enabled/Disabled	Description
		When users mouse over this menu option, the following tool-tip is displayed: “Audits the current project for errors such as components whose properties don’t match the lines they are associated with: VALIDATE”. User Assistance will be coming up with additional second level tool-tip content as part of their documentation efforts.
File > Configure Validation Settings…	Enabled	Invokes the VALIDATECONFIG command, which allows users to specify a variety of settings that effect how VALIDATE operations are performed. This option appears immediately following the new Validate Project menu option. When users mouse over this menu option, the following tool-tip is displayed: “Specify settings such as whether validation operations occur in the background and whether tray notifications are displayed when errors are encountered: VALIDATECONFIG”

4.3.1.2 Toolbars

Two new toolbar icons are added to the P&ID toolbar, as depicted in the following screenshot:

The new icons are added immediately to the right of the existing Data Manager option, without a separator. Note that the actual icons will be developed by our graphics team later in the development cycle. The following table describes these toolbar options:

ID	Tool-tip	Enabled/Disabled	Description
1	Validate Project	Varies	Invokes the VALIDATE command, which performs a validation operation on the current project. If no project is currently loaded in the Project Manager, this menu option is disabled. This menu option appears immediately following the existing Open Project menu option. When users mouse over this menu option, the abbreviated tool-tip listed in the Tool-tip column is displayed. If they continue to hover over this button, the following secondary tool-tip is displayed: “Audits the current project for errors such as components whose properties don’t match the lines they are associated with: VALIDATE”. User Assistance will be coming up with additional second level tool-tip content as part of their documentation efforts.
2	Configure Validation Settings	Enabled	Invokes the VALIDATECONFIG command, which allows users to specify a variety of settings that effect how VALIDATE operations are performed. This option appears immediately following the new

			Validate Project menu option. When users mouse over this button, the abbreviated tool-tip listed in the Tool-tip column is displayed. If they continue to hover over the control, the following secondary tool-tip is also displayed: "Specify settings such as whether validation operations occur in the background and whether tray notifications are displayed when errors are encountered: VALIDATECONFIG".

4.3.1.3 Shortcut Menus (Object and Dialog)

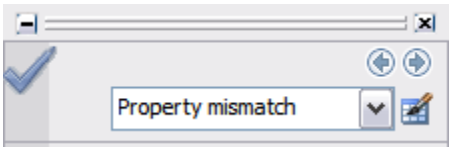
Shortcut menus for individual interface components are detailed in the sections of this specification that document the interfaces.

4.3.1.4 Double-Click Actions

Not applicable.

4.3.1.5 Dashboard Control Panels

In Cork, it is expected that various features will leverage the Dashboard infrastructure by exposing panels appropriate for interacting with P&ID drawings[18]. The Validation feature will expose a small panel, as depicted in the following screenshot.



The Validation panel will list all validation errors in the active project drawing. If a project drawing is not active in the drawing editor, or if the active drawing contains no validation errors, all controls except the P&ID Validation Settings button are disabled. If a project drawing is active with one or more validation errors, these errors are presented in a drop-down list. Users can also step through the set of errors sequentially via a pair of navigation buttons exposed through the interface.

The following table describes the controls that are proposed for the Validation panel. Additional detail about the various controls will be provided during the specification phase.

Name (also Tooltip)	Validation Errors
Tool Palette Group	N/A

[18] The final scope of Cork is still unknown at the time this CD was authored, and the original scope has been significantly curtailed over the writing of this document. A determination as to whether there is a sufficient feature set to warrant exposing P&ID Dashboard content will be made during the detailed specification phase.

ID	Control Type	Text	Enabled/Disabled	Default Value/State	Description
1	Button	Tool-tip: "P&ID Validation Settings"	Enabled	N/A	Invokes the P&ID Validation Settings dialog box.
2	Button	Tool-tip: "Previous Validation Error"	Disabled	N/A	Navigates to the previous validation error for the current drawing. This button is initially disabled until the user moves to a different navigation error. When enabled, this button sequentially steps back through each error in the list of errors from the current error. If there are no errors in the current drawing, this control is disabled.
3	Button	Tool-tip: "Next Validation Error"	Enabled	N/A	Navigates to the next validation error for the current drawing. When there are no errors in the current drawing or the last error in the list is reached, this button is disabled.
4	Drop-down list	N/A	Enabled	N/A	Lists the validation error in the active project drawing. If no validation errors are present or a non-project drawing is active, the list is empty and the control is disabled. If users use the navigation buttons to traverse the set of validation errors, the list updates with each button press to display the current validation error.
5	Button	Tool-tip: TBD during spec phase	Enabled	Toggled off	Toggle button – sets the visibility for the current error in the error list.

4.3.2 Window Interface Access

4.3.2.1 Application Status bar

As described earlier, the plot log accessible via the plot job icon exposed through the application status tray is updated to be aware of P&ID project validation errors.

4.3.2.2 Tray Icons

As described earlier in this specification (see section 4.1.1.1.1), the Validation feature by default exposes a tray icon.

4.3.3 Other Settings

4.3.3.1 Registry Settings

The last specified size and position of all dialog boxes is persisted in the registry, as well as the last-entered state/value entered into various controls.

4.4 Templates and Sample files

4.4.1 New and Modified Templates

Each P&ID template that we ship with needs to be updated to set the PIDMARKUPARC system variable as described in section 4.2.2.3.

4.4.2 New and Modified Sample Files

In order to show users how the Validation feature works, Cork should ship with at least one sample drawing file that exposes examples of each of the various error types. This sample file should be used as the basis of a tutorial provided by Tech Pubs detailing the interaction of this feature. Time and contractor resources permitting, it would be desirable to ship with multiple drawing samples based on project standards that are appropriate for various geographic locales.

4.4.3 Graphics

A variety of icon graphics are required for the Validation feature. These will be developed by a graphics consultant later in the Cork design cycle. A preliminary inventory of the expected icons required is captured in the table below. Unless otherwise indicated as existing icon, any graphics that appear below are preliminary placeholders until actual graphics are developed.

Graphic	Description
Cork application icon	14 x 14 icon. Appears in upper left corner of all application dialog boxes. To be developed later in the Cork cycle.
	15 x 16 icon - existing. Appears in the Project Manager tree. Represents a DWG file that is locked for editing by a user.
Validation error icon	15 x 16 icon. Appears in the Project Manager tree. Represents a DWG file that contains one or more validation errors. To be developed later in the Cork cycle.
New Validation Error icon	15 x 16 icon. Appears in the Validation Errors panel of the Project Manager. Represents a previously unviewed validation error.
Previously Unviewed Validation Error icon	15 x 16 icon. Appears in the Validation Errors panel of the Project Manager. Represents a previously viewed validation error.
Validation tray icon	15 x 16 icon. Appears in the system tray when background validation operations are enabled. To be developed later in the Cork cycle.

5 Constraints and Considerations

5.1 Drawing Format and Legacy Issues

5.1.1 Legacy Considerations

There is no requirement that the Validation feature work with pre-Cork drawings or projects. In order to leverage the Validation feature, users must be working with a Cork project, or a project migrated to Cork format from an earlier project format.

5.1.1.1 Round-Tripping

P&ID projects currently do not support round tripping. The Validation feature is no different in this regard.

5.2 Interaction with Other AutoCAD Features

5.2.1 Intra-Release Dependencies

5.2.1.1 Dependencies on Other Cork Features

The Cork feature has potential dependencies on the property propagation feature. The extent of these dependencies will not be known until the completion of the CD phase.

5.2.1.2 Cork Feature Dependencies on This Feature

5.2.2 Cross Feature Impact

Feature Impact	Yes/No	Description
3D Visualization	No	
Plotting	Yes	As described earlier in this document, the Plot and Publish Details dialog needs to be updated to have awareness of P&ID Validation errors. Prior to invoking plot or publish operations, it must also be determined if any of the drawings in the working set are members of the current project. If so and VALIDATECONFIG is set so the plot/publish operations invoke a VALIDATE operation, those drawings that are part of the project must be checked for validation errors.
DWF	No	
Sheet Sets	No	
Design Center	No	
Data Extraction	No	

Feature Impact	Yes/No	Description
eTransmit	No	
Other common AutoCAD commands and features	No	

5.3 Interoperability with Other Products

5.3.1 Vertical and Other Product Considerations

Vertical Name	Yes/No
AutoCAD Architecture (formerly ADT)	No
AutoCAD MEP (Mechanical, Electrical, and Plumbing) (formerly ABS)	No
AutoCAD Plant P&ID	Not applicable.
Revit	No
Inventor	No
Civil 3D	No
Plant	Yes
Mechanical Desktop	No
Impression	No
Buzzsaw	No
Constructware	No
FM Desktop	No
Manufacturing Collaboration Services	No
Viewers	No
AutoCAD Mechanical	No

Vertical Name	Yes/No
Land Development Desktop	No
Vault	No
Map 3D / MapGuide	No
3DS Max	No
Maya	No
VIZ	No
Imaging	No

5.3.1.1 Impact to Plant 3D

Many of the error conditions that are pertinent to P&ID's will also be pertinent to 3D Plant users. From a user consistency and interoperability perspective, the Plant 3D application should leverage similar work-flows and UI's for validating project drawings as appropriate.

6 Designer's Notes

6.1 Open

1. Have graphics team create icons for the Validation Settings dialog tree.

2. At feature team review, discuss whether the validation markup label should expose a grip.

6.2 Resolved

1. Need to determine if validation settings are global or per project. Resolved 7/26 – simplifies things if validation settings are global. Go with this approach unless there is compelling request from customers for a per-project approach.

2. The original Validation design exposed configuration settings that checked for the presence of a variety of AutoCAD objects. The initial Guinness implementation, however, ultimately made use of a number of these object types, most notably blocks (for title blocks in the templates as well as for certain components) and text. In light of this, design questions the usefulness of this requirement. Resolved – discussed with Derry and Jorge. We can differentiate between P&ID vs. AutoCAD blocks, and thus this requirement still seems useful. Expose functionality through Dashboard (possibly also through Validation Settings) that controls which markups are plotted/published. Ideas for settings: 1) Don't plot any markups; 2) Plot only visible markups; 3) Plot all markups. Other idea – handle via the Task Dialog that is displayed when validation errors are encountered. Resolved: handled via the Task Dialog.

3. Consider adding a new Task Dialog that shows up the first time a validation error is uncovered. Have the dialog give some background information about what a validation error is, with a link to a more detailed Help topic which describes the default settings and makes recommendations for best practices. Resolved: future directions.

7 Future Directions

1. Double-click behavior of error objects – consider using the new abbreviated property inspector control so that users can edit erroneous properties.