



**ProjectWise
Design Integration**

CONNECT Edition Update 2

A Guide to the ProjectWise Rules Engine
including Enhancements

BENTLEY SYSTEMS (UK) LTD.



Document & System History

Document Version Information

Prepared By	Graham Hirst – Bentley Systems (UK) Ltd
File Name	ProjectWise CONNECT Edition - Rules Engine Guide V1.0 (A4).docx
Date	25 May 2017
Status	For Information

Document History

Version	Date	Details	By
0.1	10 April 2017	New Draft document including enhancements introduced to ProjectWise Design Integration CONNECT edition Update 2	GPH
1.0	25 May 2017	Issued in A4 & Letter Formats	GPH

Table of Contents

Document & System History	2
Document Version Information	2
Document History	2
Table of Contents	3
1 ProjectWise Rules Engine	5
1.1 Overview	5
1.2 General Enhancements	6
1.2.1 WRE User Security	6
1.2.2 Confirmation Dialog Box	7
1.2.3 Version Creation	8
2 Workflow Rules Engine - Action Types	10
2.1 Overview	10
2.2 Action Types [ProjectWise CONNECT Edition – Update 2 and newer]	10
2.2.1 Message Box	10
2.2.2 Audit Trail Comment	11
2.3 Action Types [ProjectWise V8i SELECTseries 4 Refresh and newer]	12
2.3.1 Change State	12
2.3.2 Set Attribute	12
2.3.3 Clear Attribute	12
2.3.4 Create Version	13
2.3.5 Update Title Block	13
2.3.6 Set Final Status	13
2.3.7 Change Revision Number	14
2.3.8 Shift Revision History	14
3 Workflow Rules Engine - Condition Checks	17
3.1 Overview	17
3.2 Condition Failed Dialog	17
3.2.1 Existing Behaviour	17
3.2.2 ProjectWise CONNECT Update 2 Enhancement	17
3.3 Attribute Checks [ProjectWise CONNECT Edition – Update 2 and newer]	18

3.3.1	EQUAL_TO	18
3.3.2	NOT_EQUAL_TO	19
3.3.3	IN	20
3.3.4	NOT_IN	20
3.3.5	REGEXP_MATCH.....	21
3.4	Attribute Checks [ProjectWise V8i SELECTseries 4 Refresh and newer].....	22
3.4.1	EMPTY	22
3.4.2	NOT_EMPTY	22
3.5	Reference File Attribute Check	23
3.5.1	REFS_VALID	23
4	Configuring the Rules Engine	24
4.1	Operation Types.....	24
4.2	Rules	26
4.2.1	Action Order	27
4.3	Settings.....	28
4.3.1	Rules Engine Settings.....	28

1 ProjectWise Rules Engine

1.1 Overview

The ProjectWise Workflow Rules Engine (WRE) is a ProjectWise Explorer extension which was introduced with ProjectWise V8i (SELECTseries 4) Refresh Edition. It has been enhanced for ProjectWise CONNECT Edition – Update 2.

WRE allows ProjectWise administrators to configure document processing actions for each state in a workflow and allows the implementation of non-linear workflows. In addition, it allows the use of non-standard version strings, and the ability to increment them automatically. Finally, drawing version history attributes can be added to an environment which can be maintained automatically to update issue attributes or tags in a drawing sheet.

Workflow rules are imported through the Rules Engine node in ProjectWise Administrator. The WRE system user credentials are no longer stored in the Rules Matrix sheet, but are configured in the Rules Engine Node in ProjectWise Administrator.

Any user who has the Workflow Rules Engine extension installed in ProjectWise Explorer will be able to use the rules, assuming they also have access to the documents and workflows that are using the rules.

WRE is configured to work with specific user-assigned workflows and it overrides the standard ProjectWise workflow state-change commands and displays user-defined operations in the ProjectWise document Change State menu.

- The WRE is only enabled for specific workflows as defined by the configuration settings. Any workflow that is not enabled for WRE will behave as a default ProjectWise workflow.
- When enabled, an Operation is the user-defined command that replaces the standard ProjectWise state change commands **Next**, **Previous** and **Change**. When selected, the Operation triggers a Rule with a confirmation dialog.

NOTE: Whilst enabling the Rules Engine overrides or disables some standard ProjectWise commands such as New Version and the State Change commands, a user must have the relevant ProjectWise rights to carry out these actions for WRE to work if these actions are configured.

- A Rule is a defined set of Actions configured to run in a specific, sequential, order.
- An Action is a specific ProjectWise task that may be carried out on the document or one of its environment attributes. A library of built in actions is provided with the Rules Engine.
- One or more Condition checks may be run on the document before the actions are triggered of a specific Rule.
- Rules and their Actions are defined and managed in an Excel spread sheet (Rules Matrix) and imported into ProjectWise Administrator.

1.2 General Enhancements

1.2.1 WRE User Security

1.2.1.1 Overview

There are two ProjectWise system users that are used in conjunction with the Workflow Rules Engine:

Super User – This is the ProjectWise account used for actions marked on the Rules tab as requiring a "super user" (Super User=TRUE). This is a user with elevated rights over normal users to allow actions to be carried out if the user running the operation does not have permission to do so.

For example, with the Super User setting an attribute may be set, even though the user does not have write access to the document in the current state. When such an action occurs, Audit Trail will show the Super User as the user who carried out the action.

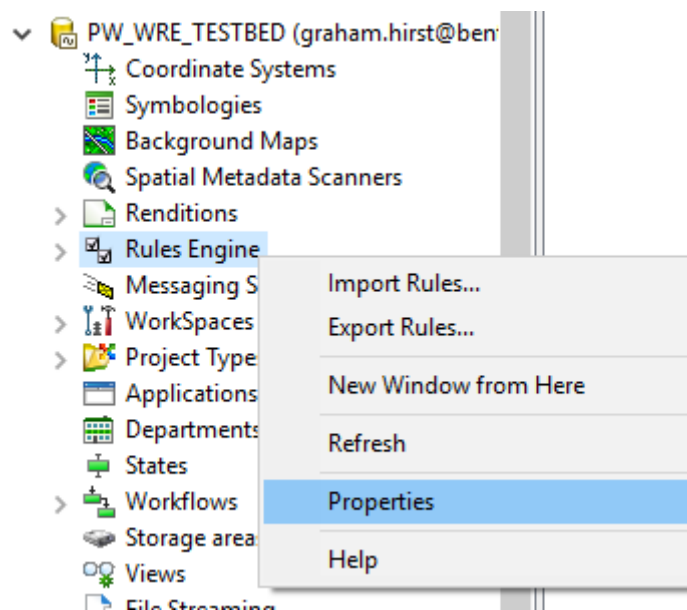
Rollback User - This is the ProjectWise account used for rolling back failed operations.

Both users must be a member of the Administrators group in ProjectWise.

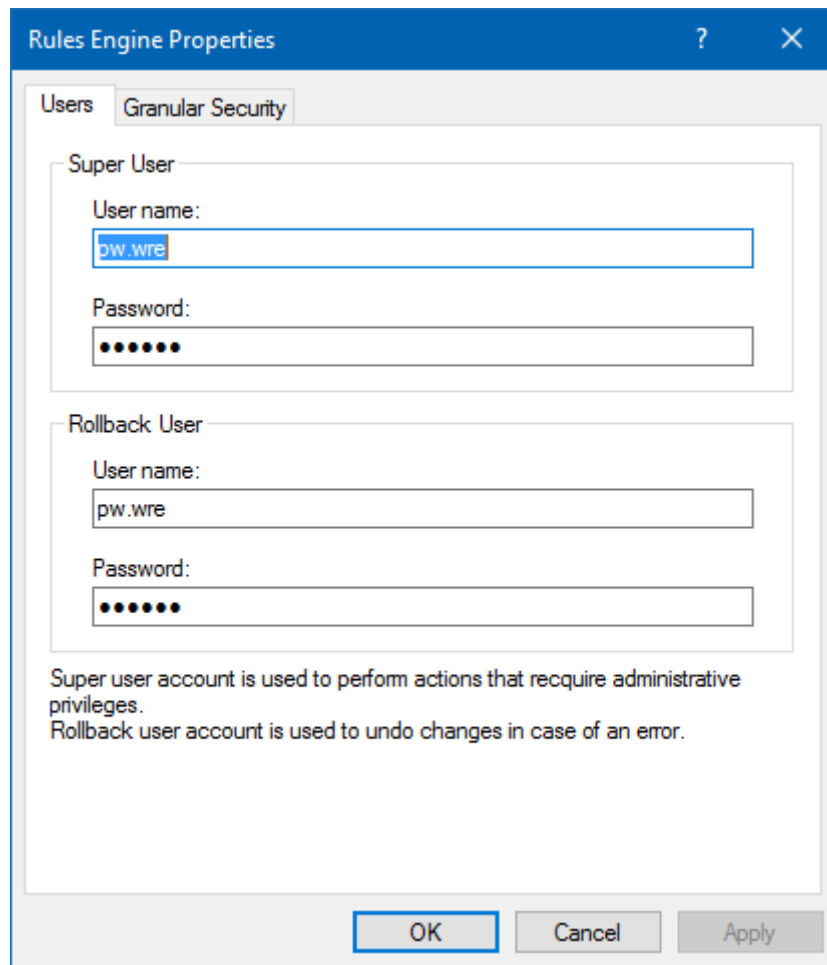
1.2.1.2 Enhancement

Previously, both user's credentials have been defined in the Settings Tab of the Rules Matrix sheet and were imported and validated during each import of the rules.

Now the users must be set in the properties of the Rules Engine node of the ProjectWise Administrator. This only needs to be done once, unless the user's credentials are changed. The username and password need to be entered for each user, and ProjectWise will validate both the password and whether the user is a member of the Administrator Group.



Select Rules Engine > Properties to add users...

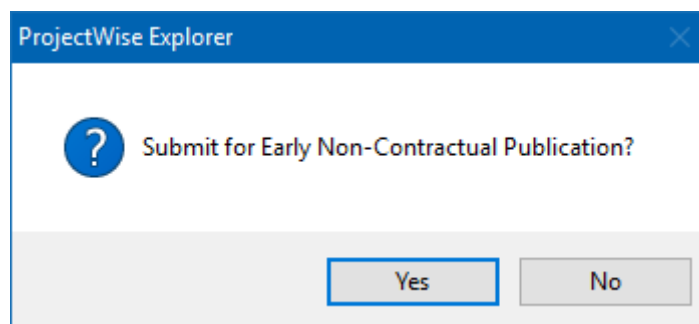


Adding Users to the Rules Engine Properties

1.2.2 Confirmation Dialog Box

1.2.2.1 Existing Behaviour

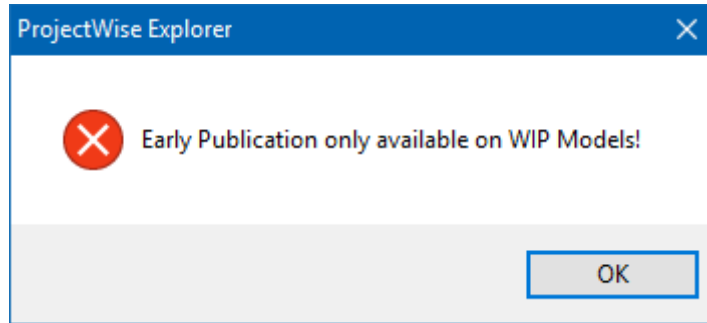
When an Operation triggers a Rule, the user is presented with a confirmation dialog box such as follow:



To carry on with the Operation, the user clicks **Yes**. To cancel the Operation and do nothing, the user selects **No**. The message presented is configurable depending on the defined operation.

1.2.2.2 Enhancement – No Actions Dialog Box

If an operation has no actions, a message dialog box is displayed with the Prompt message:



There is no longer a need to run a dummy action such as clearing an unused attribute to suppress an error message that no actions have been defined.

1.2.3 Version Creation

1.2.3.1 Current Behaviour

The default ProjectWise New > Version command is disabled for any documents that are in a workflow that is configured to use the Rules Engine. This is because typically, the version creation process involves other WRE actions such as changing document state and modifying the revision/version string.

1.2.3.2 Enhancement

It is now possible to enable Manual Versioning for one or more WRE configured workflows. This is done by means of a new setting, **MANUAL_VERSION_WORKFLOWS**, and the Value is set to the name or names of each workflow that needs manual versioning enabled.

	A	B
1	Name	Value
2	BNT_REV_NO_ATTR_NAME	RV_REV_1
3	BNT_REVISION_HISTORY_COUNT	12
4	BNT_REVISION_ATTRIBUTES	RV_REV_;RV_DATE_;RV_NOTE_;RV_BY_;RV_CHK_;RV_APP_;RV_AUTH_
5	MANUAL_VERSION_WORKFLOWS	<WORKFLOW_1;> <WORKFLOW_3;> <WORKFLOW_12>
6	RULES_MATRIX_FILE	WRE Demo V1.xlsx
7	RULES_MATRIX_DATE	07-Apr-17
8	RULES_MATRIX_MOD_BY	Graham Hirst
9		
10		
11		

Specified Workflows enabled for Manual Versioning

	A	B
1	Name	Value
2	BNT_REV_NO_ATTR_NAME	RV_REV_1
3	BNT_REVISION_HISTORY_COUNT	12
4	BNT_REVISION_ATTRIBUTES	RV_REV_;RV_DATE_;RV_NOTE_;RV_BY_;RV_CHK_;RV_APP_;RV_AUTH_
5	MANUAL_VERSION_WORKFLOWS	ALL
6	RULES_MATRIX_FILE	WRE Demo V1.xlsx
7	RULES_MATRIX_DATE	07-Apr-17
8	RULES_MATRIX_MOD_BY	Graham Hirst
9		
10		
11		

Rules Settings

All Workflows enabled for Manual Versioning

	A	B
1	Name	Value
2	BNT_REV_NO_ATTR_NAME	RV_REV_1
3	BNT_REVISION_HISTORY_COUNT	12
4	BNT_REVISION_ATTRIBUTES	RV_REV_;RV_DATE_;RV_NOTE_;RV_BY_;RV_CHK_;RV_APP_;RV_AUTH_
5	MANUAL_VERSION_WORKFLOWS	NONE
6	RULES_MATRIX_FILE	WRE Demo V1.xlsx
7	RULES_MATRIX_DATE	07-Apr-17
8	RULES_MATRIX_MOD_BY	Graham Hirst
9		
10		
11		

Rules Settings

NO Workflows enabled for Manual Versioning (Default Behavior)

2 Workflow Rules Engine - Action Types

2.1 Overview

In ProjectWise, each Workflow has a unique name. Therefore, the combination of Workflow, State, and Operation is also unique, i.e. if you select the Operation **Advance** in the first State, e.g. **Work in Progress**, of a Workflow this is a unique occurrence to which we assign a Rule. Only one combination of Workflow, State, and Operation will trigger a specific Rule.

Each Rule is assigned one or more sequential Actions. These actions are run on the document or documents the user has selected before choosing the required Operation.

A Rule is not run unless one or more optional Conditions are satisfied. These are detailed in section 3.

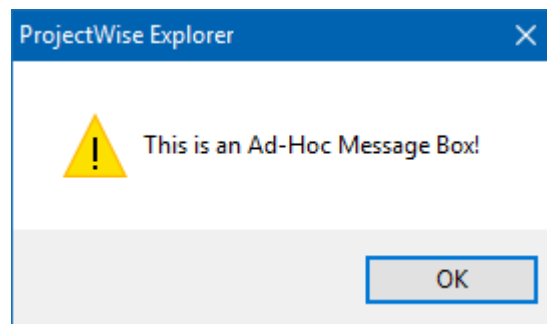
2.2 Action Types [ProjectWise CONNECT Edition – Update 2 and newer

The ProjectWise Rules Engine comes with a predefined list of Actions which can be used to make user-defined Rules. The following actions have been added to ProjectWise CONNECT Edition – Update 2.

2.2.1 Message Box

Action Type	MSG_BOX
Param1	Message to be shown
Param2	Not used

This Action shows a message box with the text specified in Param1.

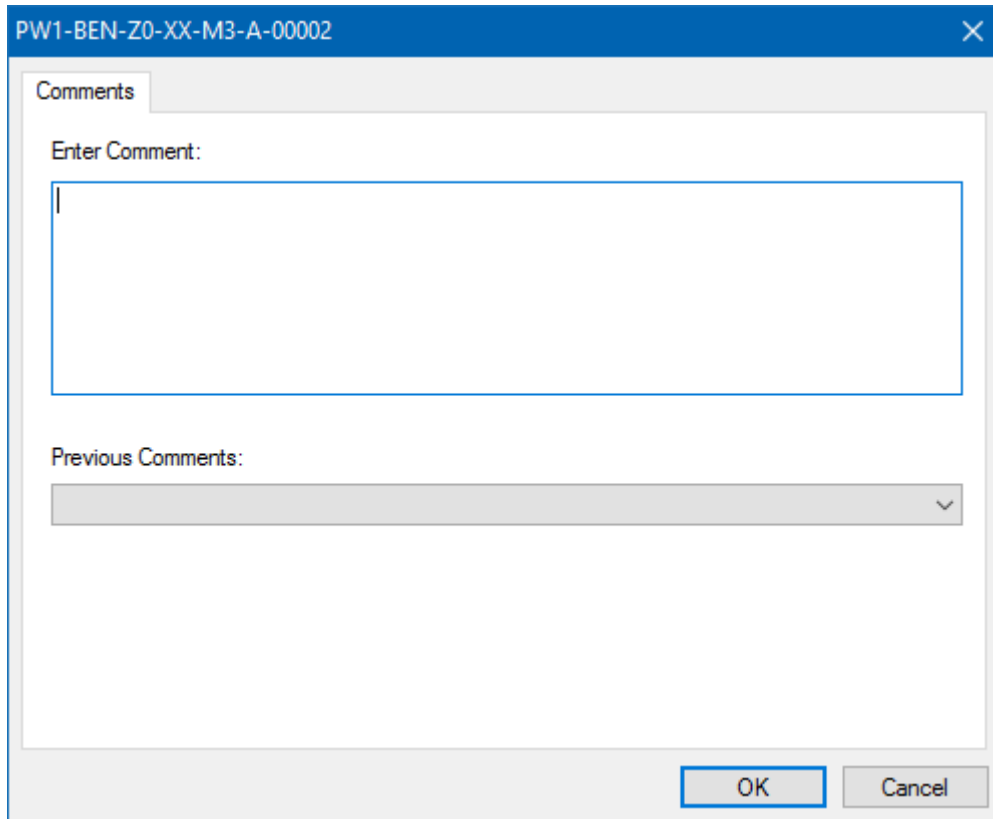


When the user clicks OK, the box closes, and the next Action in the Rule is run.

222 Audit Trail Comment

Action Type	AUDIT_COMMENT
Param1	<Target State>
Param2	<Not Used>

This action presents the audit comment prompt and saves the entered comment in the audit trail for the document processed. (Comment is added only if 'OK' button is pressed in dialog).



This action overcomes the original limitation of WRE where the standard Audit Comment dialog was suppressed when running workflows with defined rules.

2.3 Action Types [ProjectWise V8i SELECTseries 4 Refresh and newer]

2.3.1 Change State

Action Type	CHANGE_STATE
Param1	<Target State>
Param2	<Not Used>

This action is used to change the workflow state of the selected document. The target state is defined in Parameter 1. This action will invoke the ProjectWise Messaging agent rules if defined.

Non-linear State changes are supported so that states may be 'jumped' unlike default ProjectWise where the document passes through each state on the way to a new one if the Change option is used.

2.3.2 Set Attribute

Action Type	SET_ATTR
Param1	<Target Attribute Name>
Param2	<Required Attribute Value>

This action is used to set the value of an attribute of the selected document. The target attribute name is stored in Parameter 1, and the required value in Parameter 2.

2.3.3 Clear Attribute

Action Type	CLEAR_ATTR
Param1	<Target Attribute Name>
Param2	<Not Used>

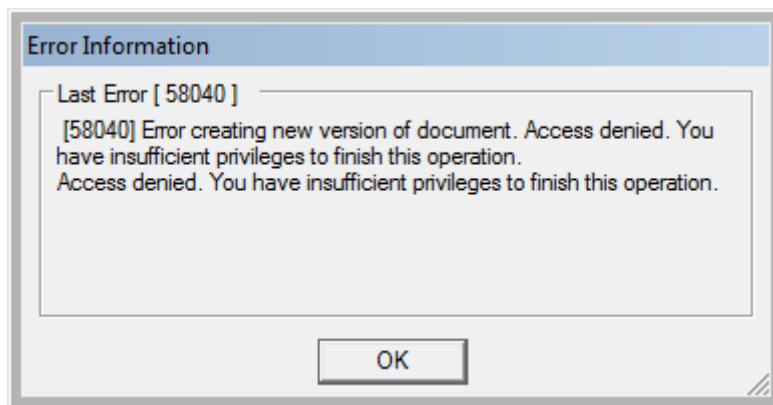
This action is used to clear the value of an attribute of the selected document. The target attribute name is stored in Parameter 1.

2.3.4 Create Version

Action Type	CREATE_VER
Param1	<Not Used>
Param2	<Not Used>

This action is used to create a new version of the selected document.

A user needs Create Version rights to run a rule with the CREATE_VER action enabled otherwise they will receive the following message:



2.3.5 Update Title Block

Action Type	UPDATE_TB
Param1	<Not Used>
Param2	<Not Used>

This action is used to silently check out and check in a cad file to update the file's embedded title block if used. If the user does not have file write access at this point the update will fail. In this case the Super User setting should be used.

UPDATE_TB relies on ProjectWise being correctly configured for Attribute Exchange.

2.3.6 Set Final Status

Action Type	SET_FINAL_STATUS
Param1	<Not Used>
Param2	<Not Used>

This action locks the selected document(s) down with Final Status. There is no action to remove Final Status as all standard ProjectWise commands are disabled for a Final Status document, including Change State which is required for WRE Operations. Documents locked with Final Status must have Final Status manually removed before any other operation can be carried out on it.

2.3.7 Change Revision Number

Action Type	CHANGE_REV_NO
Param1	<Target Revision Attribute>
Param2	<ul style="list-style-type: none"> • INC_MIN – This increases the Minor (ZZ) part of the string by 1 e.g. P01.1 > P01.2 • INC_MAJ – This increases the Major (YY) part of the revision string by 1, and resets the Minor part to 1, e.g. P01.2 > P02.1 • REMOVE_MIN - Used to remove the Minor version from the Attribute. Used for Issued documents in BS1192, e.g. P02.4 > P02

Traditionally ProjectWise has had the ability to automatically handle version strings in the form of A, B, C, etc. or 1, 2, 3, etc. Manual manipulation of the version strings was also supported.

With the introduction of the ProjectWise Rules Engine, it is now possible to handle automatic version string control. The default supported format when using the Rules Engine out of the box is **XY.Y.ZZ** where:

X = prefix letter.

YY – Numerical Revision (Major)

ZZ – Numerical Version (Minor)

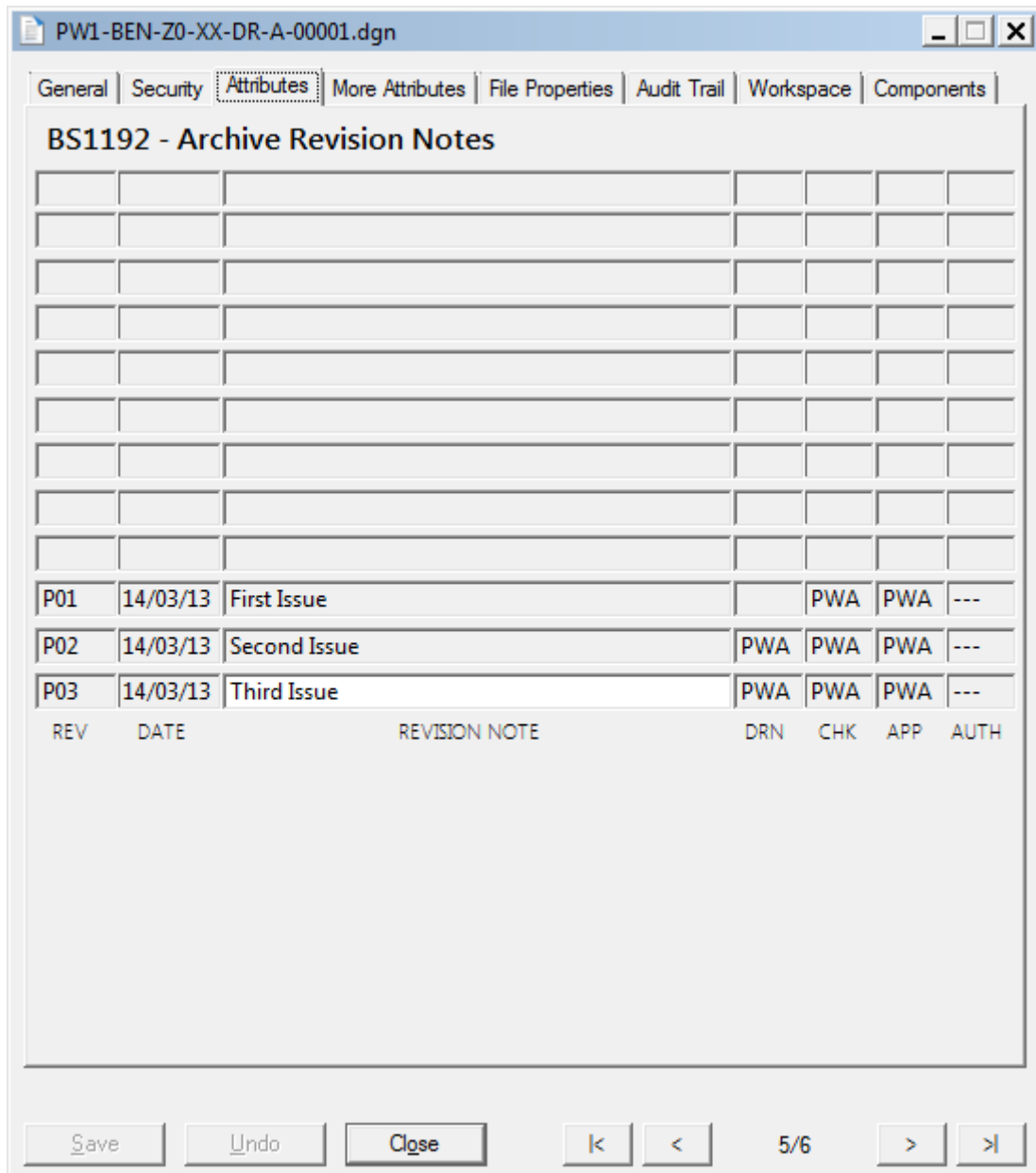
This action is used to change the Revision string of the selected document. The target attribute is defined in Parameter 1. The change action is defined in Parameter 2.

The ProjectWise native Version attribute may be synchronised with the custom revision attribute if required for any document that is in a workflow that is configured to use the Rules Engine.

2.3.8 Shift Revision History

Action Type	SHIFT_REV_HISTORY
Param1	<Not Used>
Param2	<Not Used>

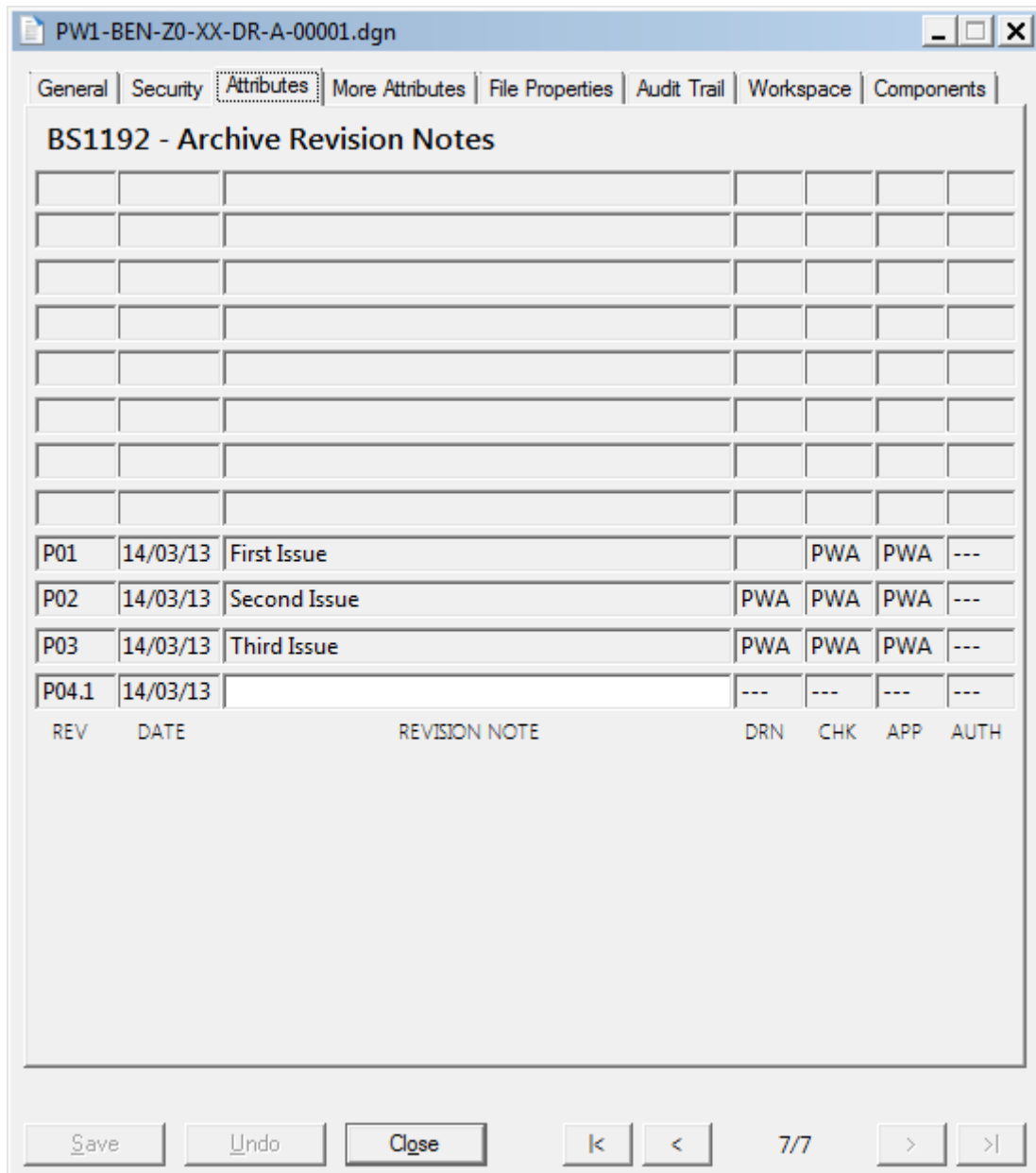
In addition to the ability to manage automated revisioning PWRE offers the ability to manage revision history attributes in the document environment. As each new version is created, the current revision attributes such as Revision, revision Note, Created By, Created Date, Approved By, etc. can be archived by moving them to Read Only attributes on the same interface. These archived attributes can be synchronized to a Revision Title block using ProjectWise Attribute Exchange rules.



ProjectWise Revision Attributes before shifting

Rev.	Date	Description	By	Chkd	App
P01	14/03/13	First Issue		PWA	PWA
P02	14/03/13	Second Issue	PWA	PWA	PWA
P03	14/03/13	Third Issue	PWA	PWA	PWA

Title Block Revision Tags/Attributes before shifting



ProjectWise Revision Attributes after shifting (and increasing MinorVersion)

P01	14/03/13	First Issue		PWA	PWA
P02	14/03/13	Second Issue	PWA	PWA	PWA
P03	14/03/13	Third Issue	PWA	PWA	PWA
P04.1	14/03/13		---	---	---
Rev.	Date	Description	By	Chkd	App

Title Block Revision Tags/Attributes after shifting (and increasing Minor Version):

3 Workflow Rules Engine - Condition Checks

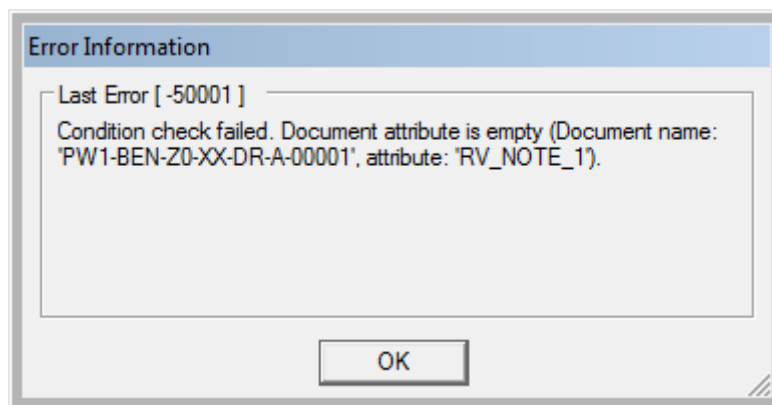
3.1 Overview

The ProjectWise Rules Engine comes with a predefined list of optional Condition Checks which may be configured for each rule. If one or more these condition checks are not met, the rule is not run, and the user will be prompted with a message describing the failed condition. Each check is separate, and until all the conditions have been satisfied, the Operation cannot be completed.

3.2 Condition Failed Dialog

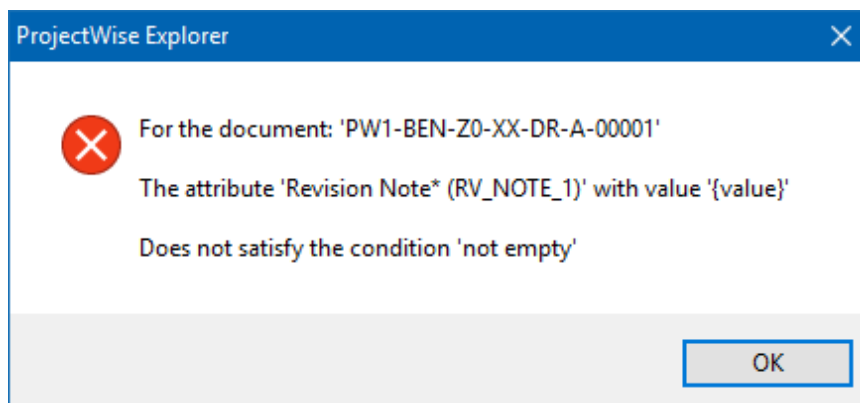
3.2.1 Existing Behavior

When a Condition Check fails, the user is presented with an Error Information box which shows the document name and the attribute affected.



3.2.2 ProjectWise CONNECT Update 2 Enhancement

A clearer Condition Failed message explains what has failed and displayed the attribute label as well as the attribute name.



3.3 Attribute Checks [ProjectWise CONNECT Edition – Update 2 and newer

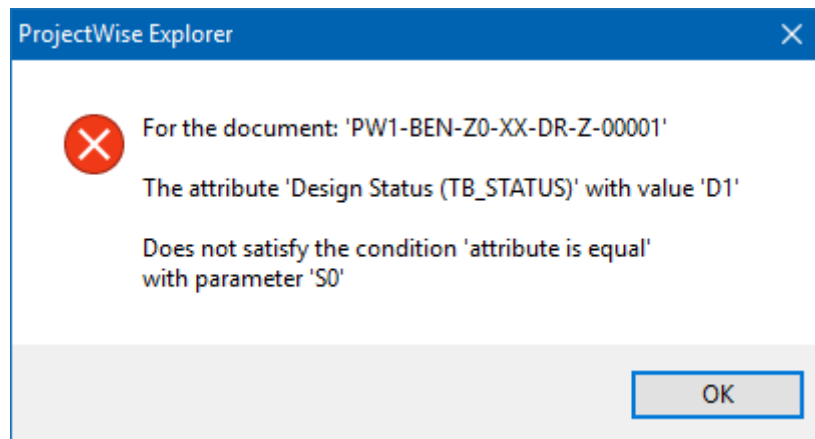
The following Condition Checks have been added to ProjectWise CONNECT Edition Update 2.

3.3.1 EQUAL_TO

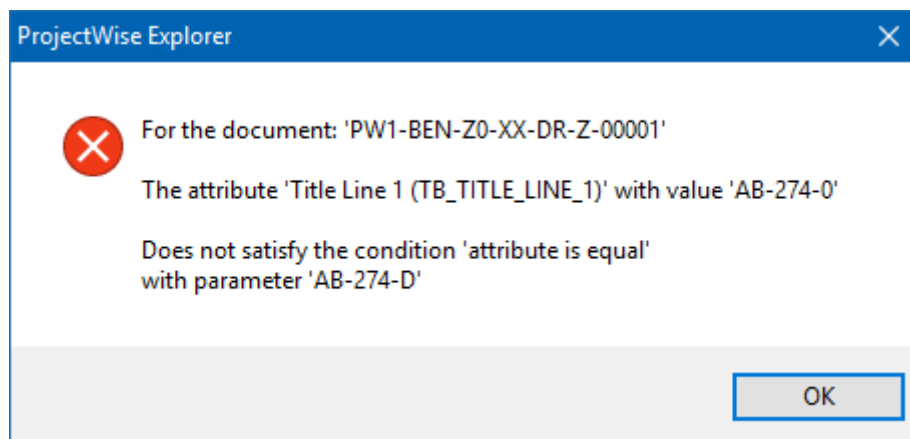
For the condition to pass, the value of the attribute in Parameter 1 MUST be equal to the static value or attribute value defined in Parameter 2, i.e. they must be the same.

When comparing attributes, the attribute name in Param2 must be of the form: \$EDIT#<Attribute Name>\$

Action Type	EQUAL_TO
Param1	<Target Attribute Name>
Param2	<Static Value> or \$EDIT#<Attribute Name>\$



EQUAL_TO - Failed Condition Message (Static Value)



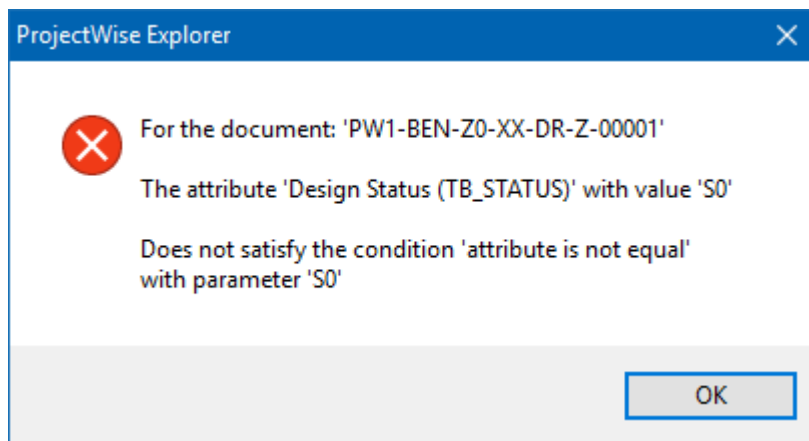
EQUAL_TO - Failed Condition Message (Attribute Comparison)

332 NOT_EQUAL_TO

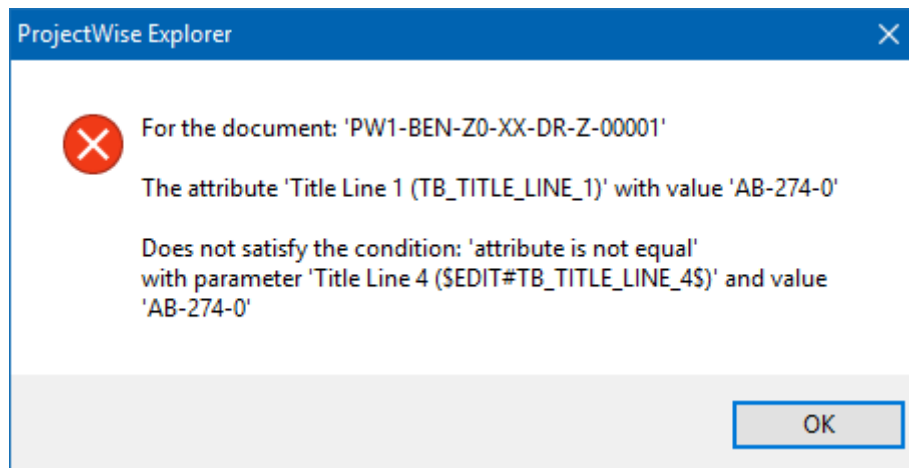
For the condition to pass, the attribute in Parameter 1 MUST NOT equal the static value or attribute value defined in Parameter 2, i.e. they cannot be the same.

When comparing attributes, the attribute name in Param2 must be of the form: \$EDIT#<Attribute Name>\$

Action Type	NOT_EQUAL_TO
Param1	<Target Attribute Name>
Param2	<Static Value> or \$EDIT#<Attribute Name>\$



NOT_EQUAL_TO - Failed Condition Message (Static Value)

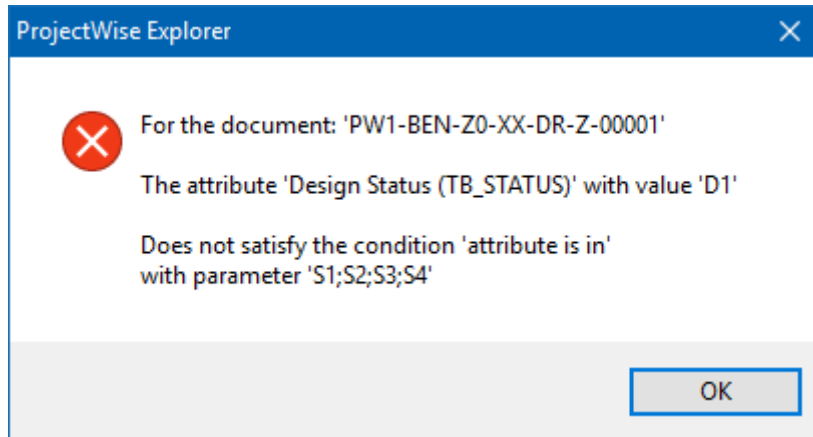


NOT_EQUAL_TO - Failed Condition Message (Attribute Comparison)

333 IN

For the condition to pass, the Attribute in Parameter 1 MUST be in the list of static values defined in Parameter 2. Each value must be separated by a semi-colon (;).

Condition Type	IN
Param1	<Target Attribute Name>
Param2	<Static Value>;<Static Value>;<Static Value>

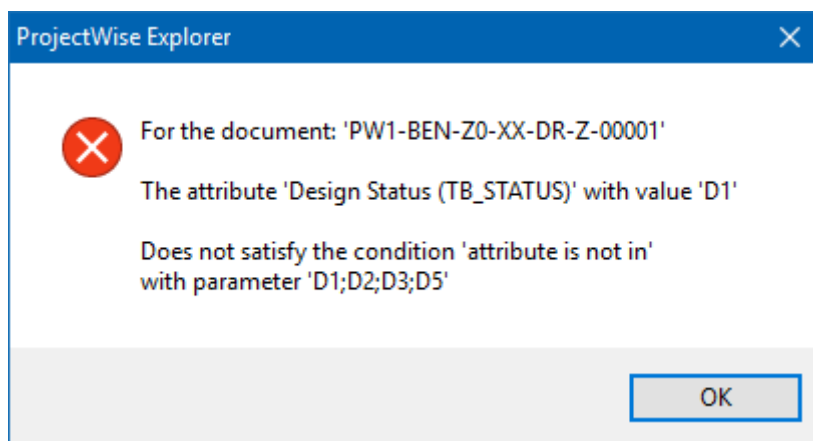


IN - Failed Condition Message

334 NOT_IN

For the condition to pass, the Attribute in Parameter 1 MUST NOT be in the list of static values defined in Parameter 2. Each value must be separated by a semi-colon (;).

Action Type	NOT_IN
Param1	<Target Attribute Name>
Param2	<Static Value>;<Static Value>;<Static Value>

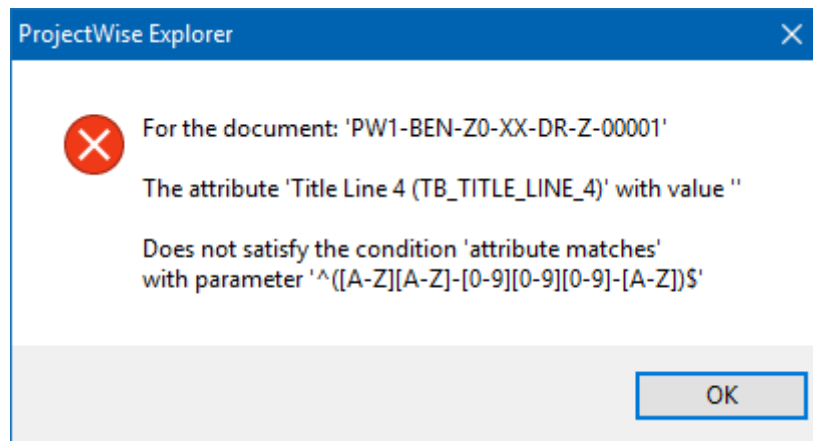


NOT_IN - Failed Condition Message

335 REGEXP_MATCH

For the condition to pass, the Attribute in Parameter 1 MUST conform to the Regular Expression (REGEXP) defined in Parameter 2. Used in situations where specific validation is strongly required. Can also be used to establish that an attribute is not empty.

Action Type	REGEXP_MATCH
Param1	<Target Attribute Name>
Param2	<Regular Expression>



REGEXP_MATCH - Failed Condition Message

For more information on Regular Expressions, please refer to:

<http://www.regular-expressions.info>.

To test Regular Expressions, please refer to:

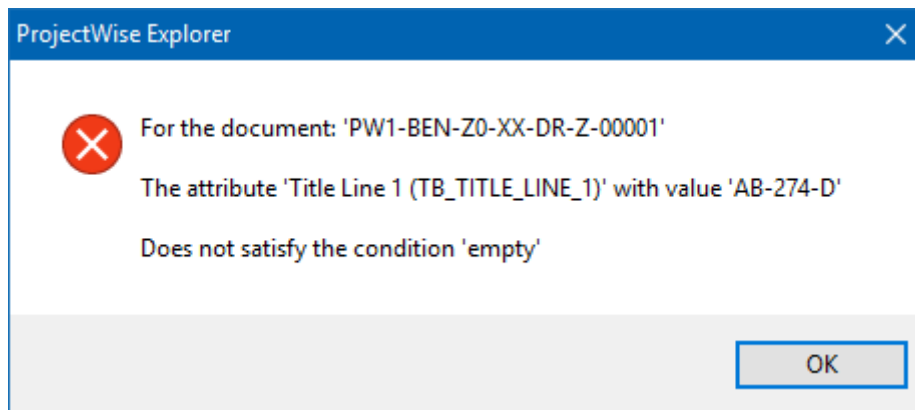
<http://www.regextester.com>

3.4 Attribute Checks [ProjectWise V8i SELECTseries 4 Refresh and newer]

3.4.1 EMPTY

The Attribute in Parameter 1 MUST be empty. The user is required to clear this attribute before the Operation can be completed.

Action Type	EMPTY
Param1	<Target Attribute Name>
Param2	<Not Used>

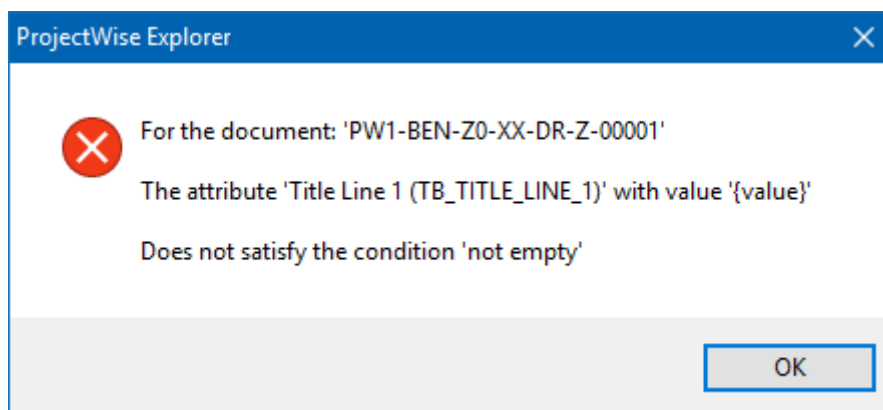


EMPTY - Failed Condition Message

3.4.2 NOT_EMPTY

The Attribute in Parameter 1 MUST NOT be empty. The user is required to fill in this attribute before the Operation can be completed.

Action Type	NOT_EMPTY
Param1	<Target Attribute Name>
Param2	<Not Used>



NOT_EMPTY - Failed Condition Message

3.5 Reference File Attribute Check

3.5.1 REFS_VALID

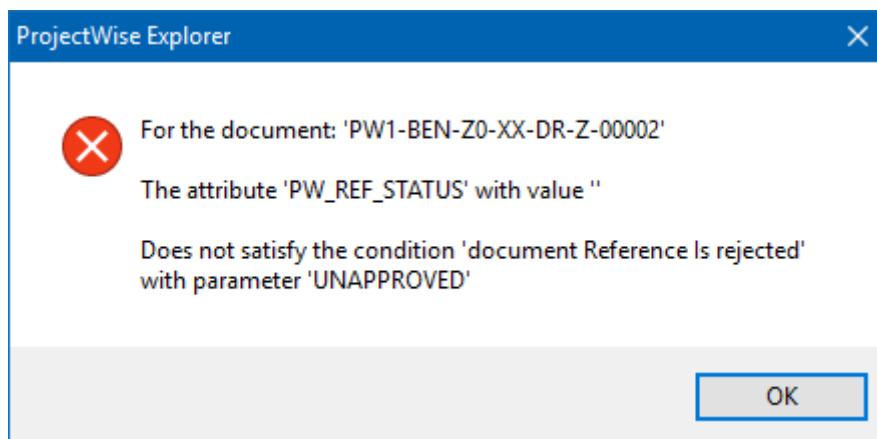
This condition check is used exclusively for CAD Files with Hierarchical Reference Sets in ProjectWise. The Attribute defined in Parameter 1 will be checked in all Reference files and if any of them contains any one of the values defined in Parameter 2, the Condition will fail, and the Master document cannot have its State changed. E.g. If any references are marked as **UNAPPROVED**, you cannot approve the Master.

As an example, as reference files are created, the default value of the reference status attribute would be set to **UNAPPROVED**.

As it is issued as an approved design, and therefore suitable to be used as a reference file with other master files, the reference status attribute could be set to **APPROVED** using the Rules Engine's **SET_ATTR** action.

Any reference files which still have **UNAPPROVED** as their reference status value would fail the **REFS_VALID** condition check, if Parameter 2 is set to **UNAPPROVED**.

Action Type	REFS_VALID
Param1	<Target Attribute Name in Reference Files>
Param2	Unapproved reference Status Value, e.g. UNAPPROVED



REFS_VALID - Failed Condition Message

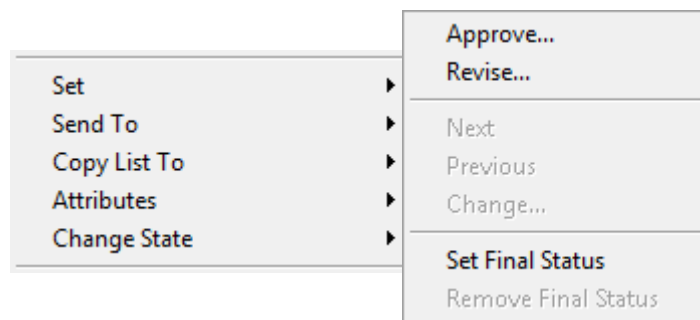
4 Configuring the Rules Engine

The ProjectWise Rules Engine is configured by means of a formatted Excel Spreadsheet, the **Rules Matrix**. There are three aspects that can be configured and these are covered on three separate Sheets in the Spreadsheet:

- Operation Types
- Rules
- Settings

4.1 Operation Types

The Operation Types settings are required to define the replacement commands that appear in the PWRE State Change Menu instead of the standard ProjectWise Commands, **Next**, **Previous**, and **Change**.



For this example, we will look at the configuration of the operations, **Approve** and **Revise**.

Each of the setting configured have their own location within the Operation Types sheet.

ID	Name	Prompt	Module	Callback	State callback	Menu order	Workflow	
							Models	Drawings
APPROVE	Approve...	Approve?	WorkflowRulesEngine.dll	Exec_DefaultCmd	State_DefaultCmd	1	TRUE	TRUE
REVISE	Revise...	Revise?	WorkflowRulesEngine.dll	Exec_DefaultCmd	State_DefaultCmd	2	TRUE	TRUE
REJECT	Reject...	Reject?	WorkflowRulesEngine.dll	Exec_DefaultCmd	State_DefaultCmd	3		

The Operation entries start on **Row 3**. Each of the Columns is detailed for each setting.

Column Name	Description
ID	This is the unique operation identifier, which maps to the document action commands in the Rules sheet. Each type can be used with multiple workflows and therefore is only defined once.
Name	This is the label that is presented to the user in the modified ProjectWise Change State menu.
Prompt	This is the prompt message that appears in the Status Bar of the ProjectWise Explorer when the Operation is selected.

Module	Dynamically linked library which implements rules engine functionality and exposes callback methods to handle menu item actions. The only value is 'WorkflowRulesEngine.dll'.
Callback	Callback method to be executed. The only value is 'Exec_DefaultCmd'.
State Callback	Callback state to be passed when the menu item event is initiated. The only value is 'State_DefaultCmd'.
Menu Order	The order the Operation items will appear in the menu when configured in the same workflow.
Workflow	<p>Each operation can be enabled for one or more workflows. Each workflow must be defined in the second row, starting at Column H. Each operation is enabled for that workflow by assigned TRUE in the relevant workflow's column.</p> <p>In the example shown above, APPROVE and REVISE have both been enabled as actions for the Models and Drawings workflow. REJECT is a valid operation, but has not be assigned to any workflow yet, so will not appear in PWRE.</p>

4.2.1 Action Order

The main Action and Condition matrix defines the number of Actions for each rule, and the order these Actions are executed in.

In the example below, the rule has six actions:

- Workflow: **Rules Engine Workflow**
- State: **Draft**
- Operation: **Approve**
 - Action 1: Set Attribute *TRIG_SUBMITTED* to 1
 - Action 2: Clear Attribute *RejectedBy*
 - Action 3: Clear Attribute *DateRejected*
 - Action 4: Clear Attribute *Remarks*
 - Action 5: Set Attribute *PercentComplete* to 50
 - Action 6: Change State to *Pending Approval*

In the Rules Matrix, this is configured as follows (Other columns and rows hidden for clarity)

				Actions						Conditions				
				Super User										
				Param2										
				Param1										
				Type										
Workflow	State	Operation Type	Prompt	ID	1	12	20	25	26	28	1	2	3	4
Rules Engine Workflow	Draft	APPROVE	Submit Document for Approve?	1	6	1	2	3	4	5	3	4	2	1

In this example, there are also 4 conditions defined to check attributes whether they are empty. If so, the operation will fail until the attributes are populated.

4.3 Settings

There are several settings that need to be defined when configuring the Rules Engine. The Settings sheet only has two columns with headings, **Name** and **Value**.

These settings are optional.

	A	B
1	Name	Value
2	BNT_REV_NO_ATTR_NAME	RV_REV_1
3	BNT_REVISION_HISTORY_COUNT	12
4	BNT_REVISION_ATTRIBUTES	RV_REV_;RV_DATE_;RV_NOTE_;RV_BY_;RV_CHK_;RV_APP_;RV_AUTH_
5	MANUAL_VERSION_WORKFLOWS	WRE Demo

The valid settings are as follows:

4.3.1 Rules Engine Settings

4.3.1.1 BNT_REV_NO_ATTR_NAME

This is the name of the main revision attribute that will be synchronized with the ProjectWise Version attribute on the General Attributes properties page.

If set, when a document is saved, the value of this attribute will be saved into the ProjectWise version attribute.

4.3.1.2 BNT_REVISION_HISTORY_COUNT

The number of revision history rows defined.

4.3.1.3 BNT_REVISION_ATTRIBUTES

These are the attributes that are affected by the SHIFT_REV_HISTORY Action. In the example interface below, there are 12 rows of attributes, with the active row at the bottom. Each time the SHIFT_REV_HISTORY action is called, the rows are all shifted up the interface, row 1 to row 2, row 2 to row 3, etc.

To do this each attribute is defined for the number of rows required. For example, looking at Column 1, the Revision column, the bottom row has an Attribute names RV_REV_1, and the second row has attribute RV_REV_2, and so on up the screen until the top left is named RV_REV_12. These are defined in the setting table by RV_REV_ without the number. The number is defined in the next setting.

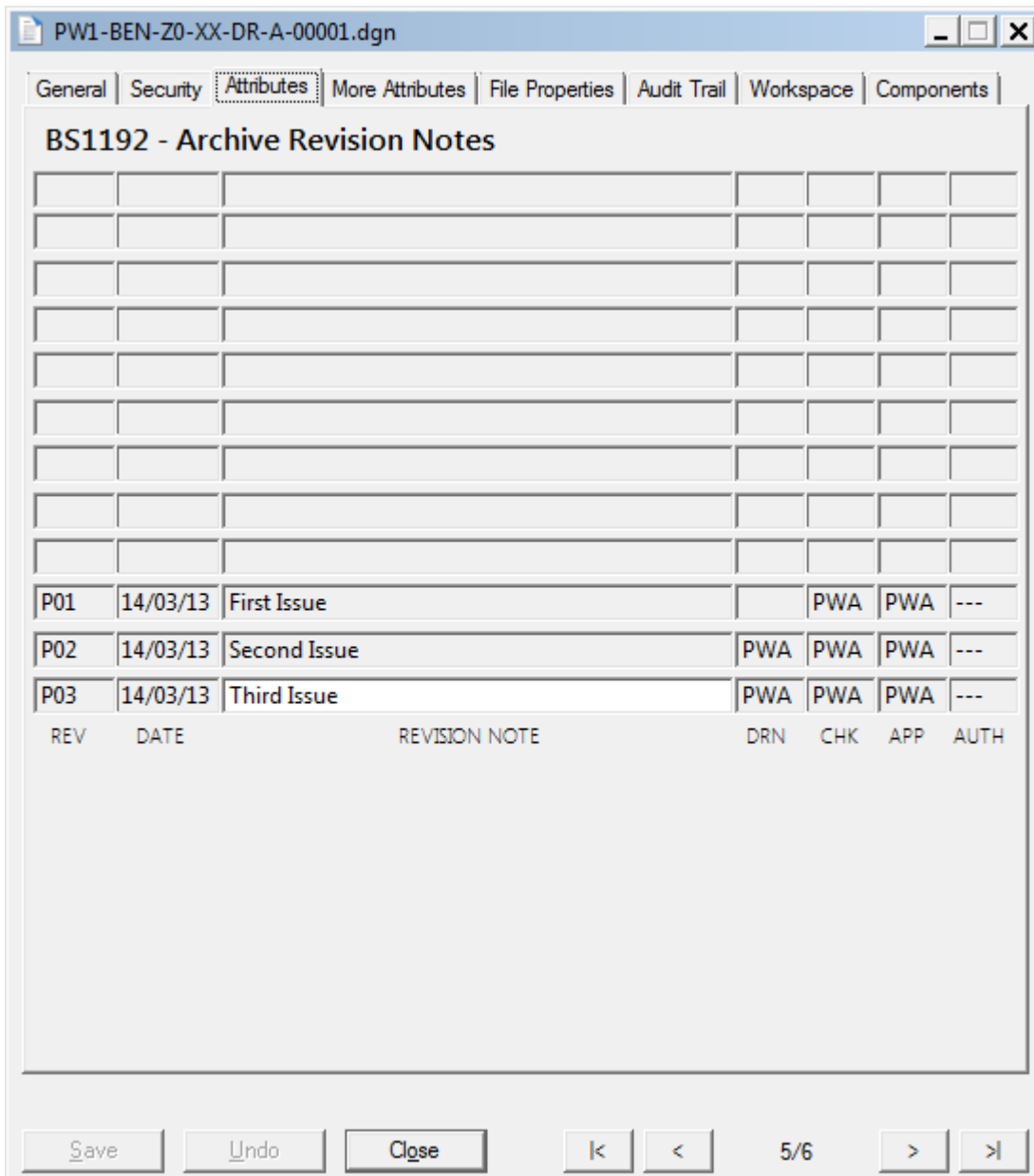
Each column follows the same pattern. So, the date column will be named RV_DATE_1 to RV_DATE_12, and will be defined in the settings as RV_DATE_.

Each value is separated by a semi-colon (;).

For the seven attributes shown in this example, these are defined in the settings as:

RV_REV_; RV_DATE_; RV_NOTE_; RV_BY_, etc.

NOTE: The attribute names are not important. Each attribute must be the same as the previous apart from the number.



4.3.1.4 MANUAL_VERSION_WORKFLOWS

See section 1.2.3.