



Rule creation and management

SKILL
LESSON

Overview

In Pega, a rule is an instruction that is automatically or manually created when configuring a Pega application.

A rule is the basic building block of a Pega Platform™ application that defines its behavior.

As rules are created, the Pega Platform™ generates application code in the background.



What is a rule

A rule is a building block of an application. There are many different rule categories and rule types in Pega, such as:

Data model

- Property
- Data transform

User interface

- Section
- Paragraph



Rule scope

When creating a rule consider reusability, context (scope), and the following:

- The type and the name of the rule.
- The branch for which the rule is related to (sprint / user story).
- The application in which the rule will be placed (Framework / Implementation).

The class determines the extent of how you can use the rule (scope in the class structure).

The ruleset identifies, stores, and manages the set of rules that define an application, a feature or a component of an application.

Type of rule

Name of the rule

Scope of the rule

How to add a new rule

1. In the Label field, enter the name of the rule, then click **Create and open**.

Create Decision Table Cancel **Create and open**

Decision Table Record Configuration

Label * **Identifier** [Edit](#)

A short description or title for this record

Context

Development branch

☐ HireMeDev
☐ HireMeTest
☒ HireMe
☐ HireMeFW
☐ UI Kit
☐ Pega Platform

Apply to * [View all](#)

Add to ruleset *

Current work item

Work item to associate
 [C](#)

How to add a new rule

1. In the Label field, enter the name of the rule, then click **Create and open**.
2. Select the **branch** from the Development branch dropdown list.

Create Decision Table Cancel Create and open

Decision Table Record Configuration

Label * Identifier [Edit](#)

A short description or title for this record

Context

Development branch

☐ HireMeDev ☐ HireMeTest ☒ HireMe ☐ HireMeFW ☐ UI Kit ☐ Pega Platform

Apply to * [View all](#)

Add to ruleset *

Current work item

Work item to associate [C](#)

How to add a new rule

1. In the Label field, enter the name of the rule, then click **Create and open**.
2. Select the **branch** from the Development branch dropdown list.
3. Select the radio button to set the application.

Create Decision Table Cancel Create and open

Decision Table Record Configuration

Label * Identifier [Edit](#)

A short description or title for this record

Context

Development branch

☐ HireMeDev
☐ HireMeTest
☒ HireMe
☐ HireMeFW
☐ UI Kit
☐ Pega Platform

Apply to * [View all](#)

Add to ruleset *

Current work item

Work item to associate [C](#)

How to add a new rule

1. In the Label field, enter the name of the rule, then click **Create and open**.
2. Select the **branch** from the Development branch dropdown list.
3. Select the radio button to set the application.
4. Click in the **Apply to** field, then press the down arrow key to select the class.

Create Decision Table Cancel Create and open

Decision Table Record Configuration

Label * Identifier [Edit](#)

A short description or title for this record

Context

Development branch

☐ HireMeDev
☐ HireMeTest
☒ HireMe
☐ HireMeFW
☐ UI Kit
☐ Pega Platform

Apply to * [View all](#)

Add to ruleset *

Current work item

Work item to associate [C](#)

How to add a new rule

1. In the Label field, enter the name of the rule, then click **Create and open**.
2. Select the **branch** from the Development branch dropdown list.
3. Select the radio button to set the application.
4. Click in the **Apply to** field, then press the down arrow key to select the class.
5. Select the dropdown list to review the Rulesets available.

Create Decision Table Cancel Create and open

Decision Table Record Configuration

Label * Identifier [Edit](#)

A short description or title for this record

Context

Development branch

☐ HireMeDev ☐ HireMeTest ☒ HireMe ☐ HireMeFW ☐ UI Kit ☐ Pega Platform

Apply to * [View all](#)

Add to ruleset *

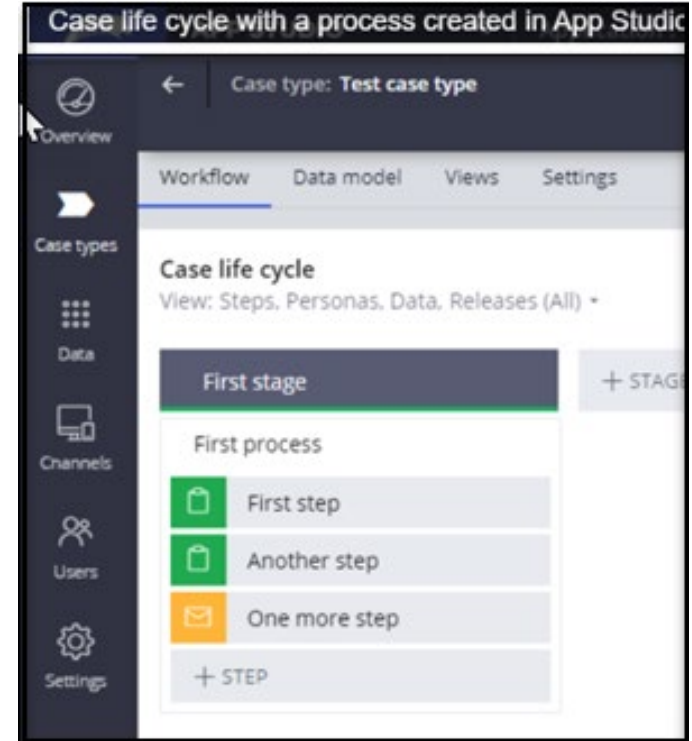
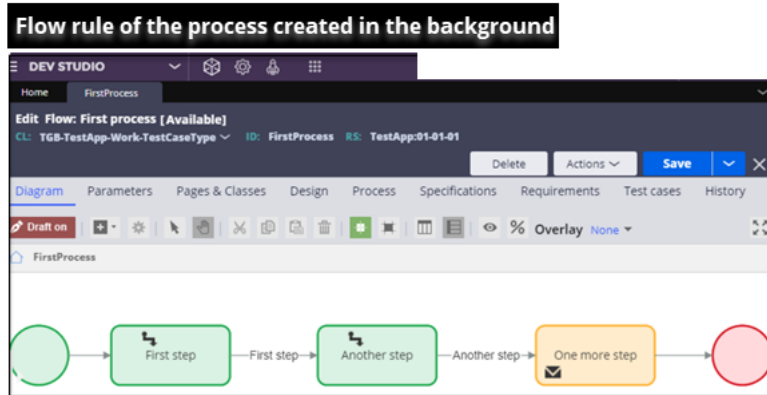
Current work item

Work item to associate [C](#)

Automated rule creation in App Studio

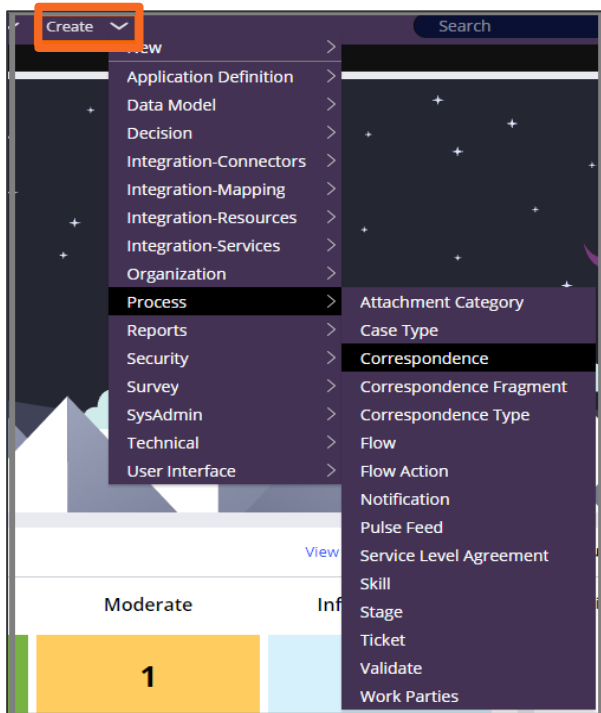
As rules are created, Pega generates application code in the background

For example: When a case type is configured in App Studio, rules that define the process flow, tasks and UI are created by the Pega Platform in the background.

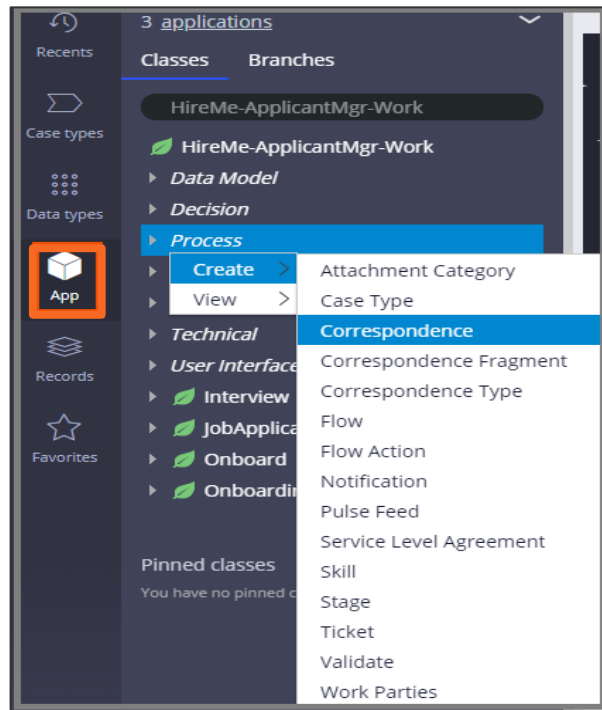


Different methods to create a rule

1. Create a rule using the Create menu.

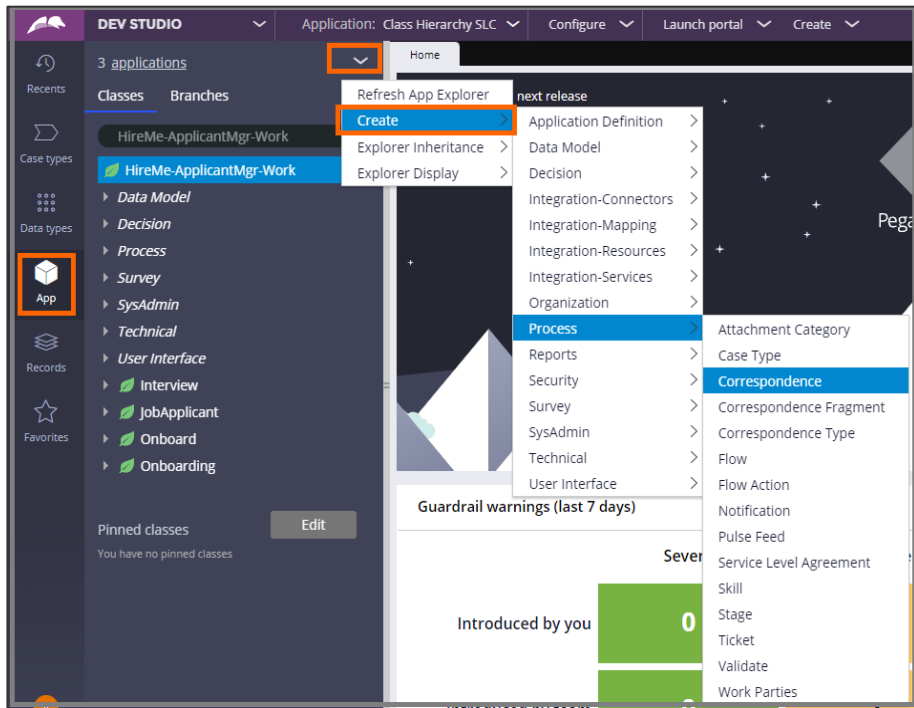


2. Create a rule from the App Explorer.

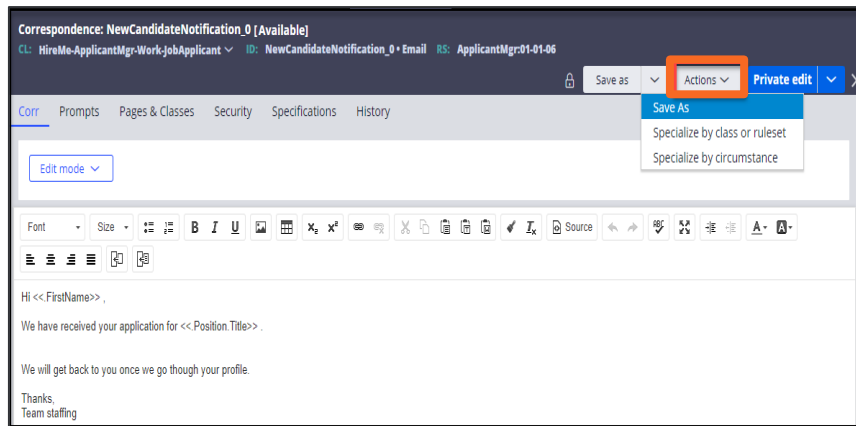


Different methods to create a rule

3. Create a rule from the App Explorer options drop down.



4. Create a rule with save as. This action takes an existing rule and applies the modifications.

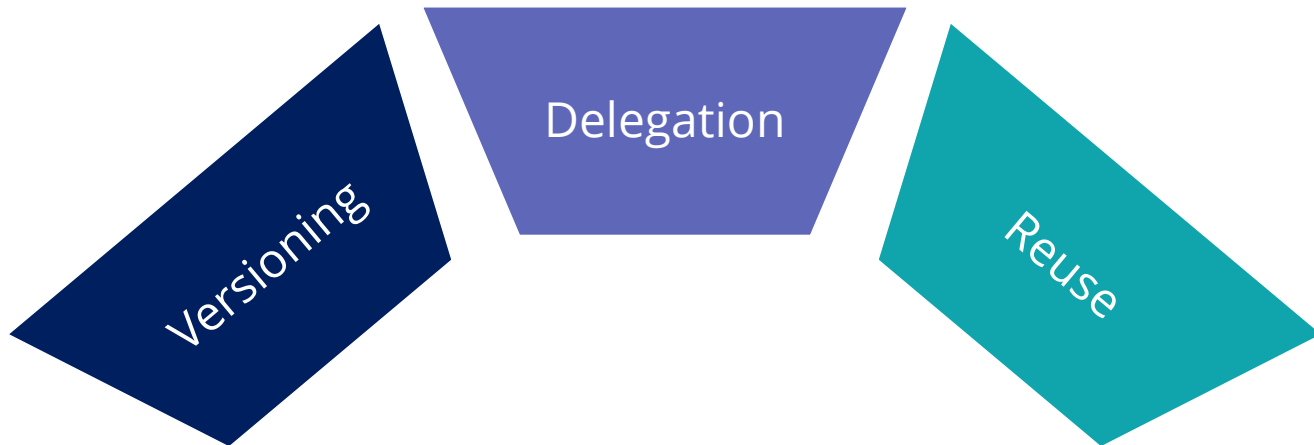


Application modularity with rules

The use of individual rules makes applications modular.

By describing case behavior with modular, task-focused rules, they can be combined and reused as needed.

This modularity provides three significant benefits: **versioning**, **delegation**, and **reuse**.



Rule versioning

Developers create a new version of a rule whenever a behavior needs to change.

Pega maintains a history of changes to a rule, allowing developers to review the change history and undo changes if needed.

Because each rule describes a specific behavior, other things remain unaffected.

Versioning

Rule: 01.01.02

Rule: 01.01.01

Delegation

Developers delegate rules to business users to allow them to update case behavior as business conditions change.

The business user updates the delegated rule, while other parts of the application remain unchanged.

Delegation

Users can modify the content of decision tables, map values, paragraphs, and correspondence rules

Reuse

Developers reuse rules whenever an application needs to incorporate an existing case behavior.



Reuse

UI Forms

Log in views

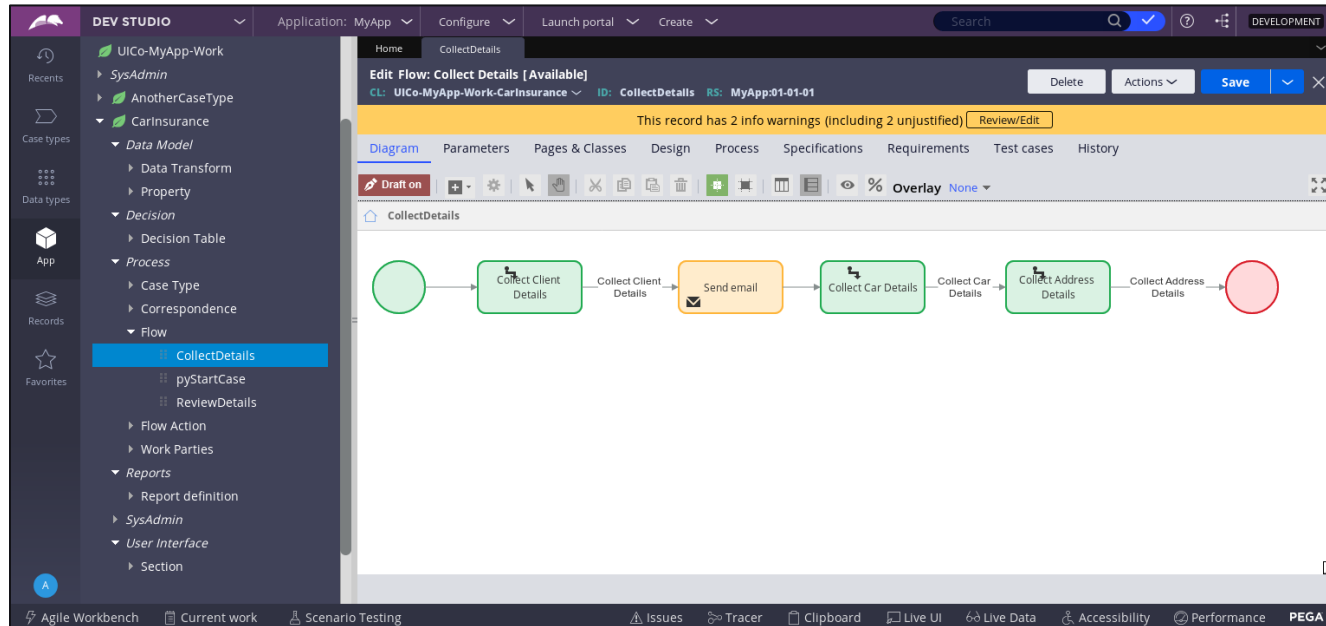
Rule updates



Overview

Checking out a rule avoids accidental rule changes or conflicts that might result from multiple developers working on the same rule.

Before editing a rule, lock it to make changes, resulting in better-quality applications.



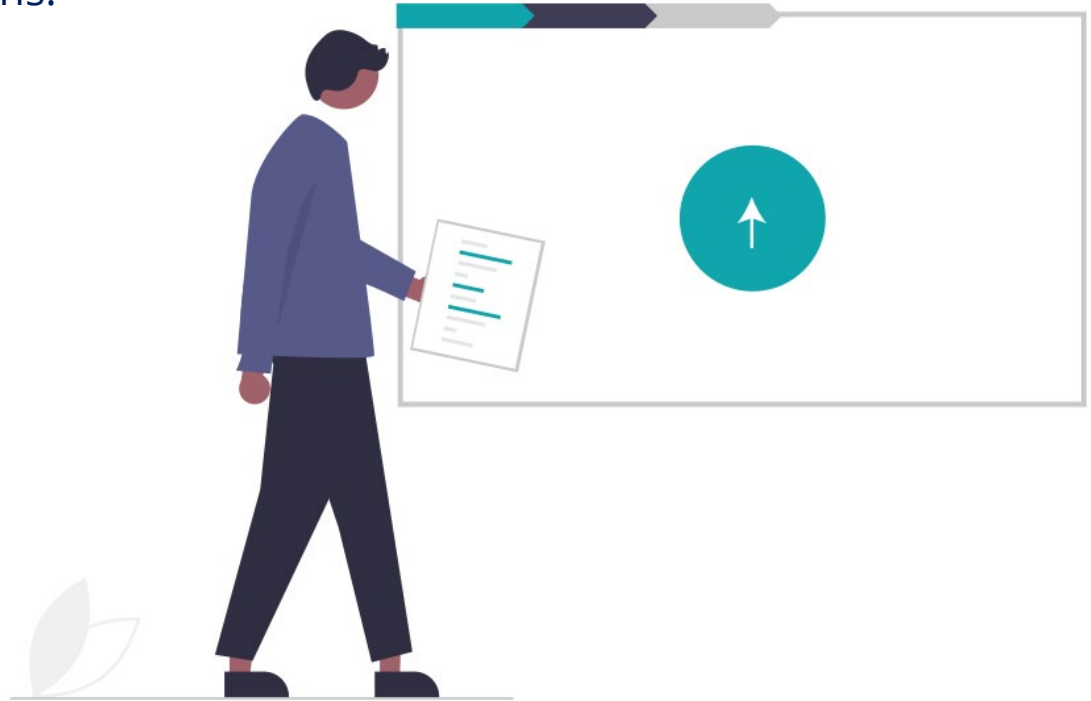


Use case

A developer's daily activity is to update rules.

Rules are updated for many reasons:

- A typo
- A business change
- A new version of the rule
- Add a missing step





Checkout

A **checkout** is a private copy of a Rule-instance that you update and later check-in to replace a base rule version.

All checked out rules reside in a personal ruleset that is only visible to the current operator.

A personal ruleset contains copies of rule instances.

- Personal rulesets are used with the checkout and check-in features.
- A personal ruleset is sometimes called a private ruleset.





Ways to checkout rules

Standard

A standard checkout button will appear when the rule belongs to a ruleset that has checkouts enabled, the operator has security privileges for checkout, and rule is not locked by another user.

Decision Table: Status Based Discount [Available]
CL: PA-HireMe-Work-PurchaseOrder ID: StatusBasedDiscount RS: HireMe [Branch: User-Story-2]

Save as Delete Actions **Check out**

This record has 2 info warnings (including 2 unjustified) View

Table Results Parameters Pages & Classes Test cases Specifications History

Select values Show conflicts Show completeness Export Import

	Conditions		Actions
	Membership	Total Amount	Return
if	Gold		→ 0.1
else if	Silver	> 500	→ 0.07
else if	Silver	> 100	→ 0.05
else if	Silver		→ 0.04
else if	Bronze	> 300	→ 0.04
else if	Bronze	> 75	→ 0.03
else if	Bronze		→ 0.02
otherwise			→ 0



Ways to checkout rules

Branch

Checkout a rule to a branch to make changes to it and then save it in the branch ruleset.

- Checking out a rule to a branch will make a copy of the rule into the branch.

Decision Table: Status Based Dicount [Available]
CL: PA-HireMe-Work-PurchaseOrder ID: StatusBasedDicount RS: HireMe:01-01-01

Save as Actions **Check out to branch**

This record has 1 info warning (including 1 unjustified) [View](#)

Table Results Parameters Pages & Classes Test cases Specifications History

Select values Show conflicts Show completeness Export Import

	Conditions		Actions
	Membership	Total Amount	Return
if	Gold		→ 0.1
else if	Silver	> 500	→ 0.07
else if	Silver	> 100	→ 0.05
else if	Silver		→ 0.04
else if	Bronze	> 300	→ 0.04
else if	Bronze	> 75	→ 0.03
else if	Bronze		→ 0.02
otherwise			→ 0



Ways to checkout rules

A **private edit** is an ability to modify a rule that is unavailable for regular checkout.

- Private edit is used when the rule is in a locked ruleset or for quick debugging without interrupting other team members.
- During a private edit, the rule is copied into an isolated sandbox.

Decision Table: Status Based Dicont [Available]
CL: PA-HireMe-Work-PurchaseOrder ID: StatusBasedDicont RS: HireMe Branch: User-Story-2

Save as Actions **Private edit**

This record has 1 info warning (including 1 unjustified) View

Table Results Parameters Pages & Classes Test cases Specifications History

Select values Show conflicts Show completeness Export

Import

	Conditions		Actions	
	Membership	Total Amount		Return
if	Gold		→	0.1
else if	Silver	> 500	→	0.07
else if	Silver	> 100	→	0.05
else if	Silver		→	0.04
else if	Bronze	> 300	→	0.04
else if	Bronze	> 75	→	0.03
else if	Bronze		→	0.02
otherwise			→	0



Checked out rules

Change history

Pega does not automatically merge rules.

The last save will overwrite previous saves.

Check-out rules that belong to an unlocked ruleset.

Checked-out rules cannot be merged and cannot be deployed.

Review the history tab of a rule to display the check-in messages that are required to check-in a rule.

Decision Table Status Based Dicoount				
Description: Status Based Dicoount				
Created on: 6/1/2020 11:18 AM by admin				
Updated on: 6/4/2020 12:56 PM by author				
HISTORY OF DECISION TABLE STATUS BASED DICOOUNT				
Time	Memo	Save Action	Updated By	Compare
6/4/2020 12:56 PM	Action performed on Instance: Save	Save	author@hiremedev	<input type="checkbox"/>
6/4/2020 12:56 PM	Another step beyond madness	CheckIn	author@hiremedev	<input type="checkbox"/>
6/4/2020 12:55 PM	Action performed on Instance: Save	Save	author@hiremedev	<input type="checkbox"/>
6/4/2020 12:55 PM	Action performed on Instance: Save	Save	admin@hiremedev	<input type="checkbox"/>
6/4/2020 12:55 PM	crazy change	CheckIn	admin@hiremedev	<input type="checkbox"/>
6/4/2020 12:25 PM	Action performed on Instance: Save	Save	admin@hiremedev	<input type="checkbox"/>
6/4/2020 12:14 PM	Action performed on Instance: Save	Save	admin@hiremedev	<input type="checkbox"/>
6/1/2020 11:18 AM	Action performed on Instance: Save	Save	admin@hiremedev	<input type="checkbox"/>
6/1/2020 11:18 AM	Action performed on Instance: New	New	admin@hiremedev	<input type="checkbox"/>
6/1/2020 11:18 AM	Copied from PA-HIREME-WORK- PURCHASEORDER!STATUSBASEDDICOOUNT in HireMe_Branch_User-Story-1 01-01-01	SaveAs	admin@hiremedev	<input type="checkbox"/>
ADD MEMO				
<div></div>				
<div></div>				
<div>Add Memo History for All Versions Close Compare</div>				



Locked rules

A checked-out rule is **locked**, and is not available for others to edit.

Users attempting to edit a checked-out rule will receive a display message with the operator ID of the person with the rule checked out and the time lapsed since checkout.

Decision Table: Status Based Discount [Available]

CL: PA-HireMe-Work-PurchaseOrder ID: StatusBasedDiscount RS: HireMe [Branch: User-Story-2]

Save as Actions Private edit

This record has 1 info This record was checked out by admin@hiremedev 8 minutes ago. [View checked out record](#).

Table Results Parameters Pa

Select values Show conflicts Show completeness Export

Import

	Conditions		Actions
	Membership	Total Amount	Return
if	Gold		→ 0.1
else if	Silver	> 500	→ 0.07
else if	Silver	> 100	→ 0.05
else if	Silver		→ 0.04
else if	Bronze	> 300	→ 0.04
else if	Bronze	> 75	→ 0.03
else if	Bronze		→ 0.02
otherwise			→ 0



Checked-out indicator

When the current operator has a rule checked out, the App Explorer will display a checkmark in a green circle next to the rule.

Checked-out rules can be found by clicking on the blue check mark at the top right of Dev Studio.

Bulk check-in can also be done here.

The screenshot shows the Pega Dev Studio interface. On the left, the 'Recents' list displays several rules, with a green checkmark icon next to the 'Status Based Dico...' rule. In the top right corner of the Dev Studio window, a blue checkmark icon in a circle is highlighted with an orange box. The main workspace displays the 'Edit Decision Table: Status Based Dico...' with a table of conditions and actions.

Conditions		Actions
Membership	Total Amount	Return
if	Gold	→ 0.1
else if	Silver	→ 0.07
else if	Silver	→ 0.05
else if	Silver	→ 0.04
else if	Bronze	→ 0.04
else if	Bronze	→ 0.03
else if	Bronze	→ 0.02
otherwise		→ 0



Check-in rules

Check-in replaces a rule version with a copy of a Rule- instance that you checked out and updated.

Comments are required when checking in rules. Provide a detailed and concise explanation of changes for a historical record.

Edit Decision Table: Status Based Discount [Available]
CL: PA-HireMe-Work-PurchaseOrder ID: StatusBasedDiscount RS: HireMe [Branch: User-Story-2]

Check in Discard Actions Save

Table Results Parameters Pages & Classes Test cases Specifications History

Select values Show conflicts Show completeness Export Import

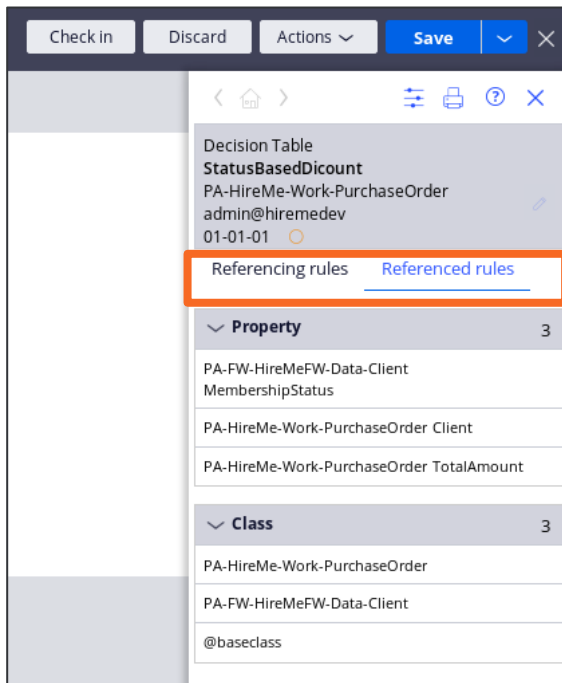
Conditions			Actions
	Membership	Total Amount	Return
if	Gold		→ 0.1
else if	Silver	> 500	→ 0.07
else if	Silver	> 100	→ 0.05
else if	Silver		→ 0.04
else if	Bronze	> 300	→ 0.04
else if	Bronze	> 75	→ 0.03
else if	Bronze		→ 0.02
otherwise			→ 0



Rule actions

While editing a checked-out rule, select **Actions** to review the following:

1. **References** display all associations to a rule, either reference to or reference by.

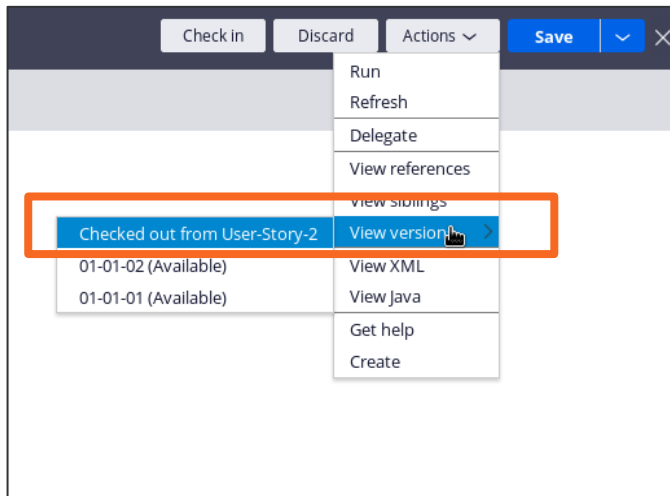




Rule actions

While editing a checked-out rule, select **Actions** to review the following:

2. **View version** - Versions allow opening previous versions of the rule.





Rule actions

While editing a checked-out rule, select **Actions** to review the following:

3. **Siblings** displays the other classes and other rulesets where the rule exist and displays the current version checked-out.

Rule-Declare-DecisionTable rules related to PA-HireMe-Work-PurchaseOrderSTATUSBASEDDICOUNT

Applies To: PA-HireMe-Work-PurchaseOrder [Get Related Rules](#)

5 RULE-DECLARE-DECISIONTABLE RULES NAMED ISTATUSBASEDDICOUNT WERE FOUND.

PA-HireMe-Work-PurchaseOrder	⊙	⊙	⊙		⊙
PA-HireMe-Work				⊙	
Applies To	HireMe	HireMe	HireMe_Branch_User-Story-2	HireMeDev_Branch_User-Story-2	admin@hiremedev
	01-01-01	01-01-02	01-01-01	01-01-01	01-01-01
	RuleSet & Version				

LEGEND

Current Class, RuleSet Version, or Rule {#} Multiple Rules

Availability
⊙ Yes ○ No ➔ Final ⊗ Blocked 🚫 Withdrawn

DETAILS



Best practices

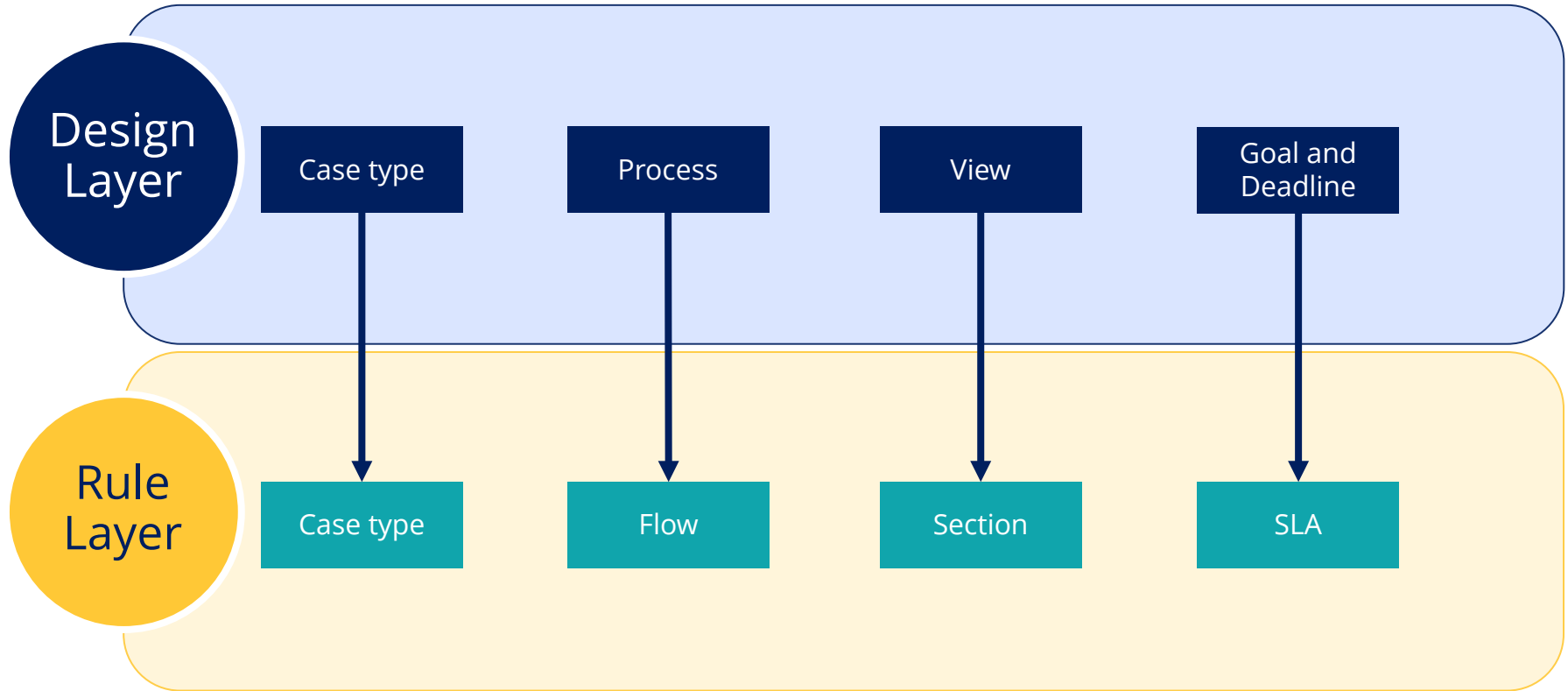
- Create a new rule from an existing or previous version.
- Check-out rules and use a personal ruleset to not impact colleagues.
- Use branching even for a small feature or improvement.
- Provide descriptive and concise comments when checking in a rule.
- Click the Checked-out records icon in the Dev Studio header to view all checked out rules.
- Use the Application Development landing page to view all checked-out rules. (Private edit rules do not appear in this list.)
- Use the restore feature to undo changes made by a checked-in version.
- Use the lock icon to see who has a specific rule instance checked out.
- Select the View version > Checked out from [version number] option in the Actions menu to view the base version of a rule checked out.

Best practices

- Check for rules in the current project, other projects, and OOTB.
- Use the Save as option and apply modifications.
- Always think about the context before creating a rule.
- Assure that the class is created in the right scope.
- Give the rule a meaningful name.
- Class name should not end in “Details”, “Info”, or “Information” – that is what they do.
- Unit test every rule created.
- Add descriptive comments to describe purpose.
- Recognize what is a Rule.
- Understand how to create a rule.
- Understand rule scope.



Schematics



Skill mastery

You understand:

- Creating rules
- Rule checkout
- Rule check-in
- Locked rules
- Private edit of a rule
- Rule actions



PEGA[®]