



Rulesets

SKILL
LESSON

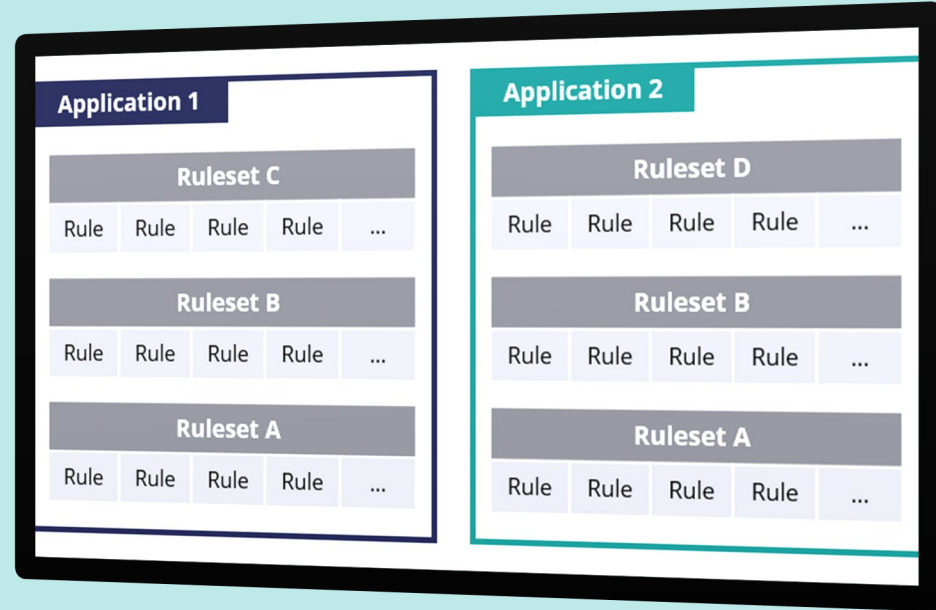
Overview

A ruleset identifies, stores, and manages the set of rules that define an application or a significant portion of an application.

To package rules for distribution as part of an application, rules are collected into a group called a ruleset.

This module covers:

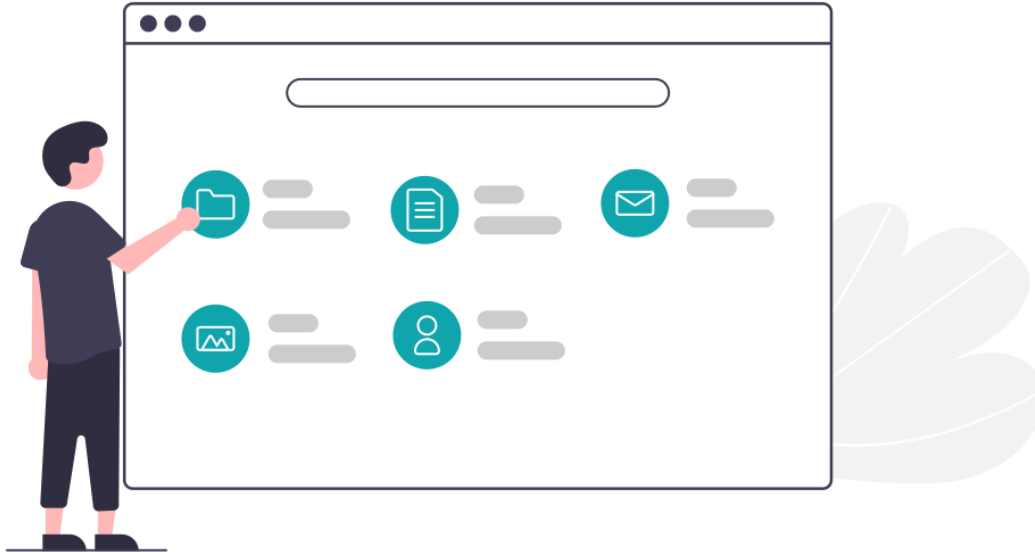
- Rulesets
 - Stacking
 - Versioning
 - Skimming



Use case

Use rulesets to group rules for accessibility and deployment.

For example: If you create a ruleset to store service-level agreement (SLA) rules, you can reuse the ruleset in many applications that process cases requiring the same SLAs.



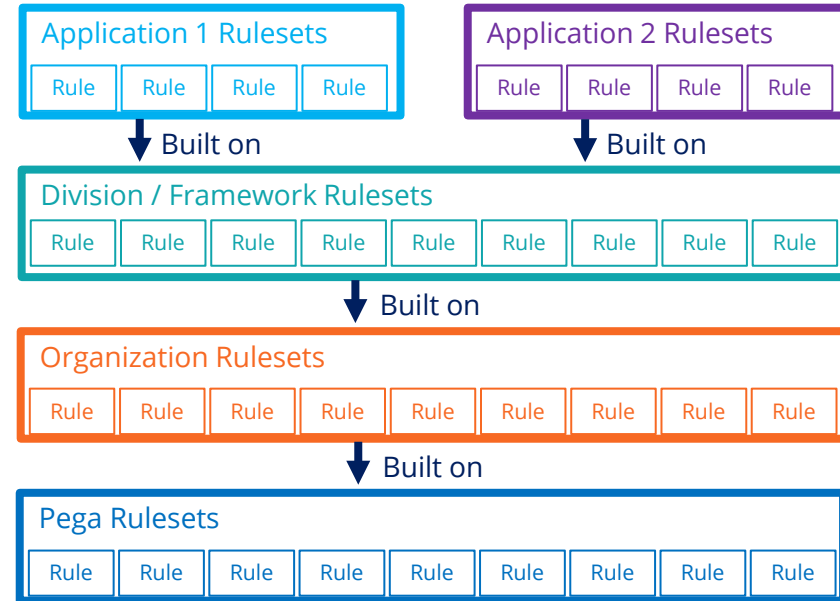
Rulesets

Create rulesets as needed for an application. The default is two rulesets for each ECS layer (org, div, framework, implementation).

Rulesets store, categorize, and version rules in an application.

They're used for packaging and deploying rules and versioning.

- Rulesets are reusable in different applications.
- They allow you to gather many rules from different classes into one ruleset that represents a feature or a component of an application.



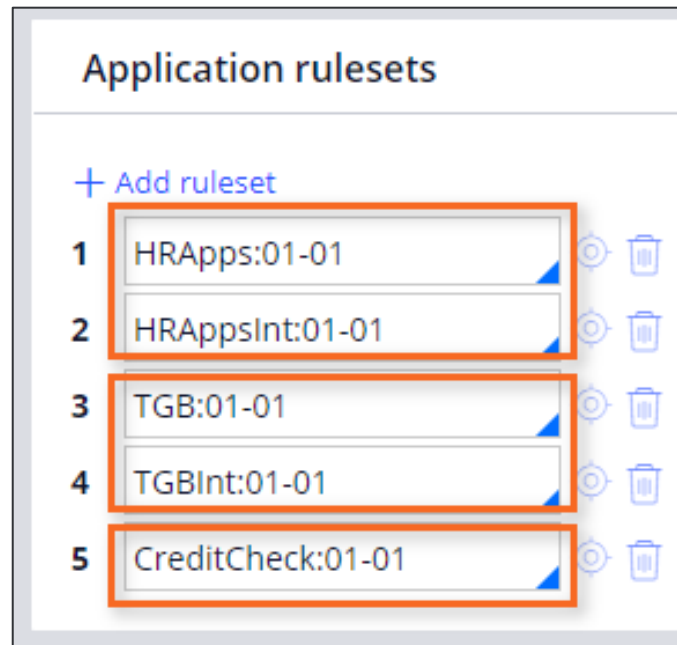


Rulesets created by the application wizard

The new application wizard creates four initial rulesets.

- Application rulesets which are specific to the application.
 - HRApps
 - HRAppsInt
- Organizational rulesets which are global to the application.
 - TGB
 - TGBInt


Newly created rules can go into the application ruleset and rules for reuse can be placed in the organization ruleset.

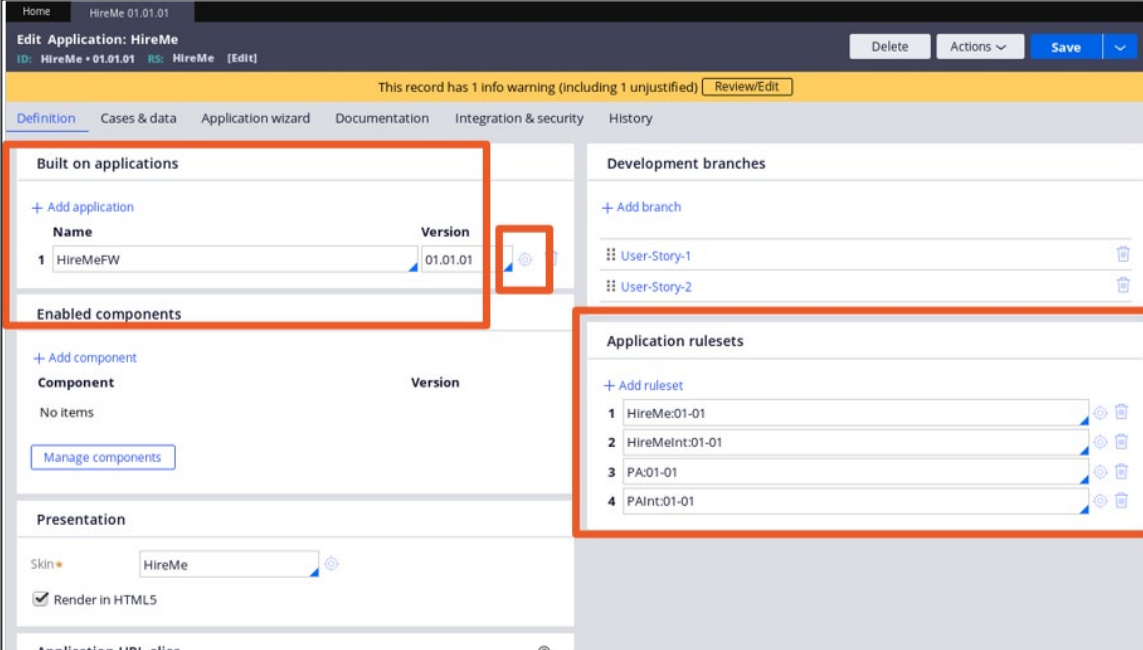




Application rulesets









Dev Studio

1. To open the application rule, click **Application > Definition**.
2. Set the Built on application and the application ruleset of an application.
3. Select the version of the ruleset to use.
4. Open the ruleset: click  next to Version.



The screenshot shows the 'Edit Application' interface in Pega Dev Studio. The application is named 'HireMe' with ID 'HireMe-01.01.01'. The 'Definition' tab is active. The 'Built on applications' section shows 'HireMeFW' with version '01.01.01'. The 'Enabled components' section is empty. The 'Presentation' section shows 'Skin' set to 'HireMe' and 'Render in HTML5' checked. The 'Application rulesets' section on the right lists four rulesets: 'HireMe:01-01', 'HireMeInt:01-01', 'PA:01-01', and 'PAInt:01-01'. Red boxes highlight the 'Built on applications' table, the gear icon next to the version, and the 'Application rulesets' list.

Built on applications	
Name	Version
1 HireMeFW	01.01.01

Application rulesets	
+ Add ruleset	
1 HireMe:01-01	 
2 HireMeInt:01-01	 
3 PA:01-01	 
4 PAlnt:01-01	 



Production rulesets

Production rulesets have at least one unlocked ruleset version in the production environment.

Production rulesets include rules that are updated in the production environment.



- Commonly used for delegated rules.

✓ Advanced

+ Add production ruleset


Production rulesets (customization)

1



☒ Place properties on thread page only (5-4 or later)

Log off redirection

Show Log off screen ▼ 



Ruleset stack

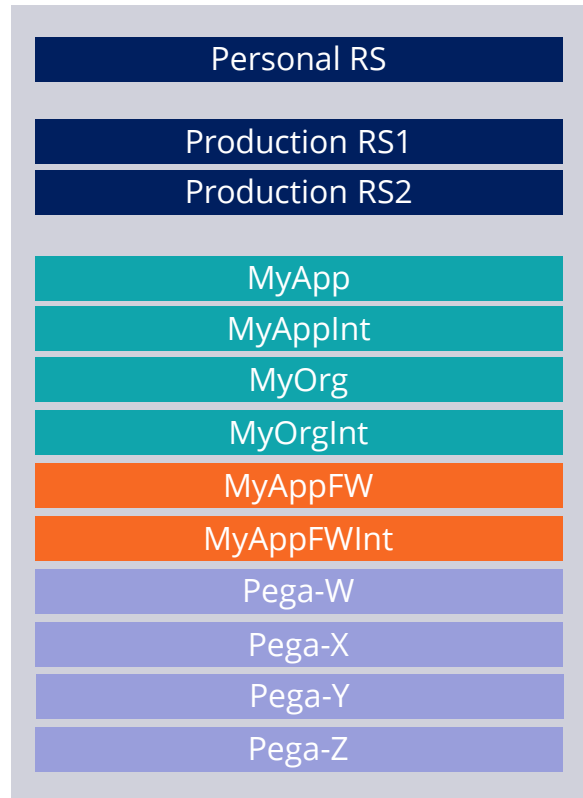
The ruleset stack lists the rulesets and ruleset versions that make up the application including those inherited from any built-on applications.

The stack corresponds to all the rulesets or rules you have access to during designing time when implanting case types and at runtime when executing a case.

As part of the stack, and on top of it, there are two types of other rulesets, personal and production rulesets.

- A personal ruleset is specific to the current operator and contains all the rules that the operator has checked out or private edit.

A production ruleset are rulesets which are unlocked in production.





Operator ruleset stack

Dev Studio

- Click the OperatorID's avatar.
- Click profile.

The operator profile opens and displays the operator details and ruleset stack.

	Name	admin	Organization	PA
	Position		Division	Div
	ID	admin@hiremedev	Unit	Unit
	Phone		Team	default@PA
Skills		Access		
Name		Rating	Application	HireMeDev 01.01.01
			Access group	HireMeDev:Administrators
			Portal Layout	Developer
			Work Pool	PA-HireMe-Work
Time		Server		
Calendar	USDefault	Name	pega	Directory /webwb
Time Zone	America/New_York	Node	localhost	URI /prweb/app/HireMe
Locale settings				
Current Locale	en_US			
Roles		Rulesets		
HireMeDev:Administrator		admin@hiremedev:		
PegaRULES:SecurityAdministrator		HireMeDev:01-01		
HireMeDev:PegaAPI		HireMeDevInt:01-01		
		PA:01-01		
		PAInt:01-01		
		HireMeTest_Branch_User-Story-1:01-01-01		
		HireMeTest:01-01		
		HireMeTestInt:01-01		
		HireMe_Branch_User-Story-1:01-01-01		
		HireMe_Branch_User-Story-2:01-01-01		
		HireMe:01-01		
		HireMeInt:01-01		
		HireMeFW:01-01		
		HireMeFWInt:01-01		
		UI-Kit-7:15-01-01		
		Pega-ProcessCommander:08-04		
		Pega-DeploymentDefaults:08-04		
		Pega-DecisionScience:08-04		

Ruleset versioning

Developers collect rules into an instance of a ruleset, called a **ruleset version**.

To update the contents of the ruleset, create a new ruleset version.

Ruleset versioning helps developers identify and manage changes to an application.

The **major version** change encompasses extensive changes to application functionality

The **patch version** consists of fixes to address bugs in an application.

Expense:

01

Major
Version

02

Minor
Version

03

Patch
Version

The **minor version** represents an interim release or enhancements to a major release.



Skimming

Skimming is saving the highest version of a rule into a new, higher ruleset version.

Skimming simplifies rule resolution and minimizes the rule data that is provided to a different version of an application.

During a minor skim, rules are stored in a higher minor version, and during a major skim, rules are stored in a higher major version.

Key to skimming - start at a major version and skim all minor and patch numbers into a new version, or you can start at some minor version and work up from there.

V 02-01-01	A	B	C	D	E
------------	---	---	---	---	---

V 01-02-02				D	
------------	--	--	--	---	--

V 01-02-01	A		C		E
------------	---	--	---	--	---

V 01-01-02		B			E
------------	--	---	--	--	---

V 01-01-01	A		C		
------------	---	--	---	--	--



Ruleset maintenance

The Refactoring wizard is for managing the reusable rulesets in an application.

System: RefactorRefresh×

[Classes](#) [Rules](#) [Rulesets](#)

Refactor Rulesets Utilities

[Copy/Merge RuleSet](#) Helps you copy RuleSets and RuleSet versions into new versions or merge several RuleSet versions into one version.

[Delete a RuleSet](#) Delete an entire RuleSet or a specific version of a RuleSet.

[Skim a RuleSet](#) Copy the highest (major or minor) version of existing rules to a higher version.

[Rulebase Compare](#) Identify differences in the rules present on two PRPC systems.

DevStudio > Configure > System > Refactor

Skill mastery

You understand:

- Rulesets
 - Stacking
 - Versioning
 - Skimming



PEGA[®]