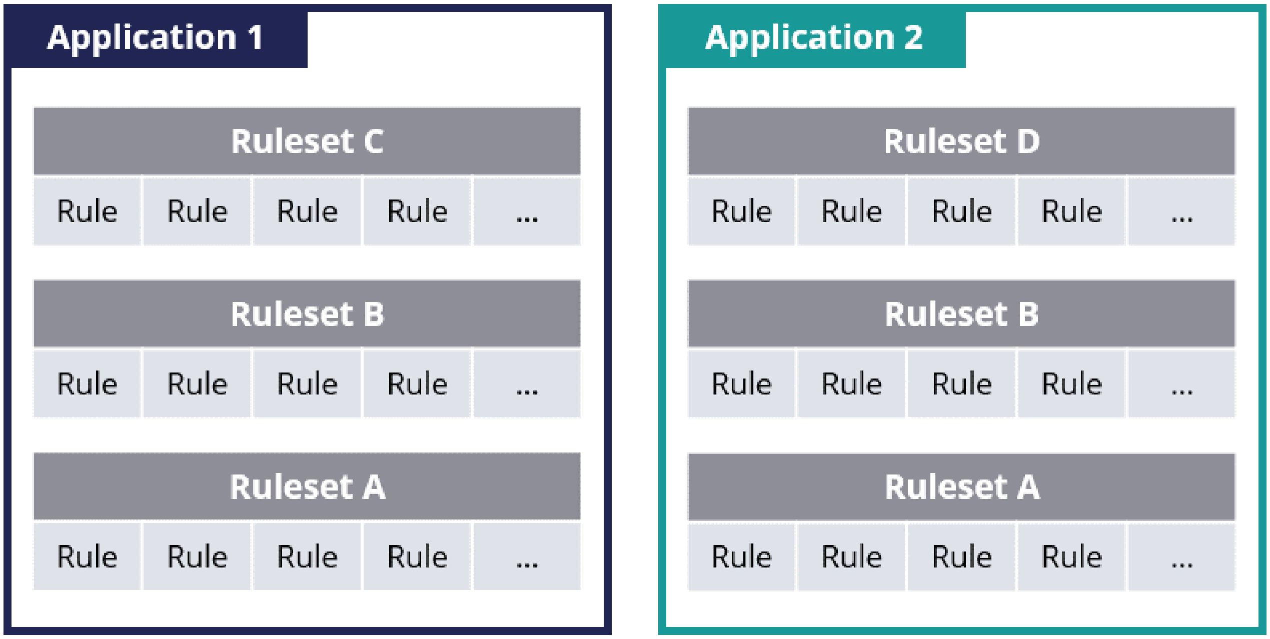
Rulesets

Rulesets

To package rules for distribution as part of an application, you collect rules into a group called a**ruleset**. A ruleset identifies, stores, and manages the set of rules that define an application or a significant portion of an application. If a rule is similar to a song, a ruleset is similar to an entire album.

Just as you can copy the album to share with a friend and allow your friend to listen to your favorite song, you can share a ruleset between applications to allow several applications to use the same rules. The ability to re-use already-built rules saves development time and effort.



Ruleset versioning

Developers collect individual rules into an instance of a ruleset, called a**ruleset version**. To update the contents of the ruleset, you create a new ruleset version. The new ruleset version is available — or unlocked — for the developer to make rule updates and add new rules. It is a best practice to lock older versions of the ruleset. Ruleset versioning helps developers identify and manage changes to an application.

> **Note:** Pega Platform™ manages the creation of rules and identifies the ruleset and version in which to store rules for applications created in App Studio.

Pega identifies rulesets by names and version numbers. For example, an application to process expense reports includes a ruleset named *Expense*. Pega appends the version number to the ruleset name.*Expense:01-02-03* is version number 01-02-03 of the *Expense* ruleset.

The version number is divided into three segments: a **major version**, a **minor version**, and a **patch version**. Each segment is a two-digit number starting at 01 and increasing to 99. Ruleset version numbering starts at 01-01-01, and increments upward.

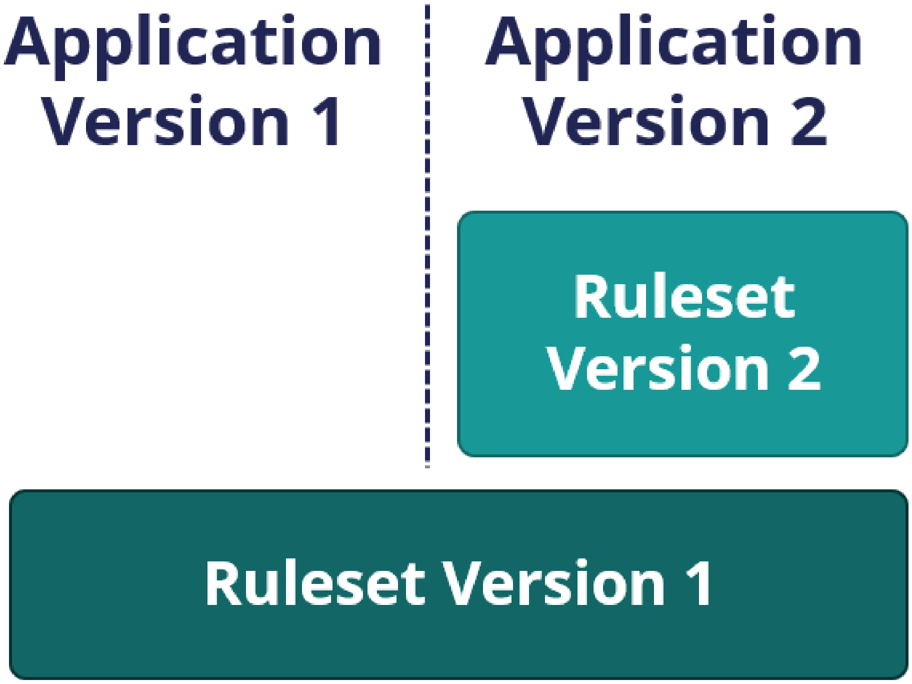
This learning is interactive and cannot be experienced offline. Please visit https://academy.pega.com to complete.

Ruleset stack

Each application consists of a sequence of rulesets called a**ruleset stack**. The ruleset stack determines the order in which Pega Platform looks through rulesets to find the rule that is in use. Each entry in the ruleset stack represents all the versions of the specified ruleset, starting with the listed version and working down to the lowest minor and patch version for the specified major version.

Each version of an application contains a unique ruleset stack. The ruleset stack allows an updated application to reference new ruleset versions that contain updates and new features. For example, Bob is a developer working on the first version of an application to manage expense reports. Bob creates rules for the first version of the application, such as processes, UIs, and notifications. Bob collects these rules into the first version of the Expense ruleset, *Expense:01-01-01*.

Months later, Tanya receives an enhancement request to update a UI in the application to collect extra information from employees due to a policy change. This update enhances the rules created earlier by Bob. Tanya creates rules to model this new behavior in a second version of the ruleset, *Expense:01-02-01*. She then uses the*Expense:01-02-01* ruleset in the updated expense reporting application.



Employees who use the first version of the application view the UI that Bob created. Only employees who use the updated application view the UI that Tanya created. The ruleset stack for the first version does not include Tanya's version, which allows users to use the first version of the application while the second version is in development.

This learning is interactive and cannot be experienced offline. Please visit https://academy.pega.com to complete.