Dunnington Software Development Incorporated

Moving At The Pace Of Integrity!

SSA TOPICS BLUEPRINT STUDY GUIDE

SSA 24.2 CERTIFICATION TOPICS

DUNNINGTON SOFTWARE DEVELOPMENT INCORPORATED

Dunnington Software Development Incorporated

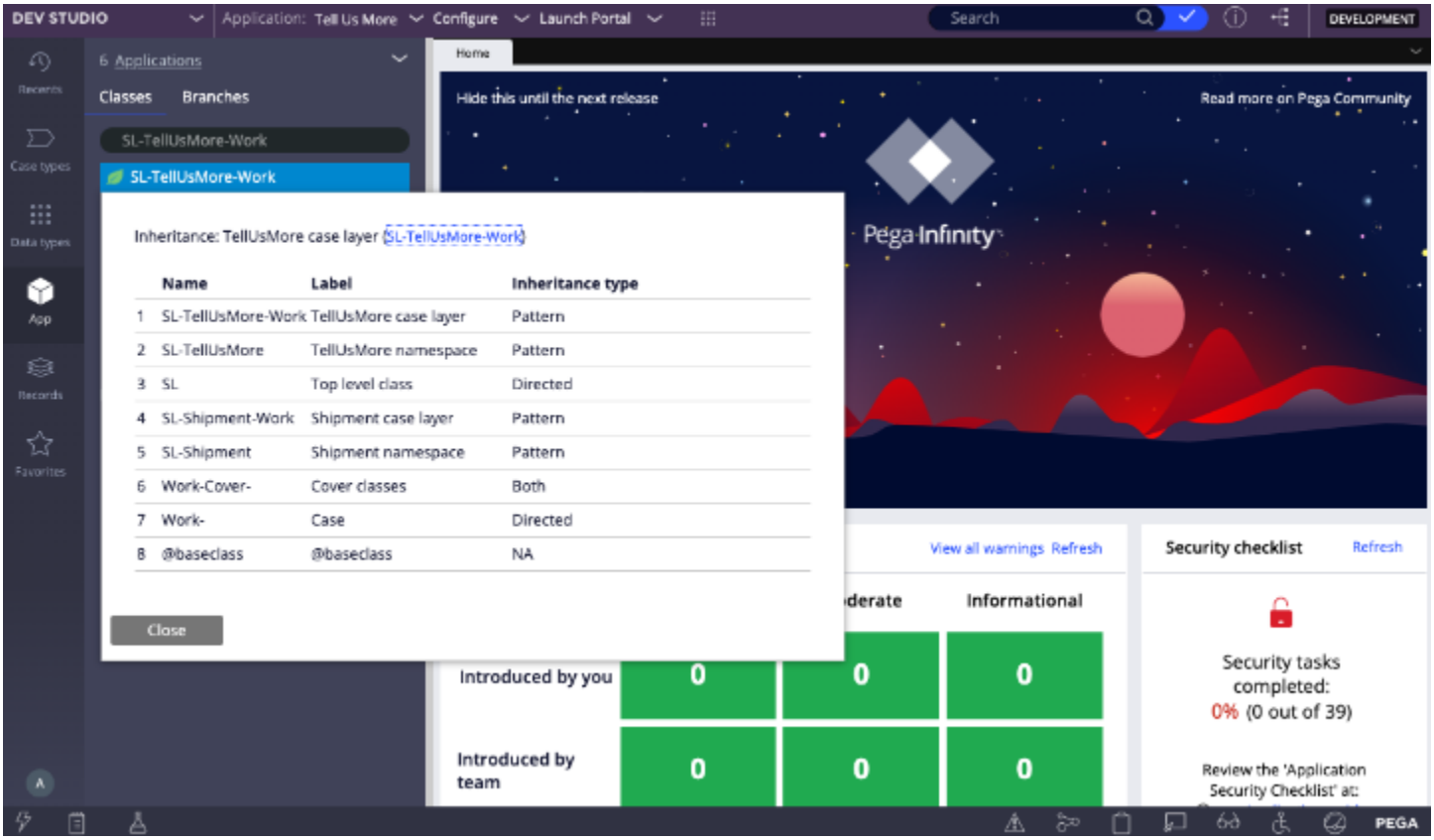
Moving At The Pace Of Integrity!

Question: What is the Enterprise Class Structure, and how does it promote rule reuse and polymorphism?

DEFINITION/DESCRIPTION(S): The **Enterprise Class Structure (ECS)** is a layered hierarchy of classes in Pega that is a core part of the Pega Platform's object-oriented architecture. It promotes **polymorphism** by allowing a single rule to be created in a generic class and then reused in more specific child classes. This reuse is achieved through **inheritance**, where child classes can inherit and specialize rules defined in their parent classes. The ECS includes four key layers:

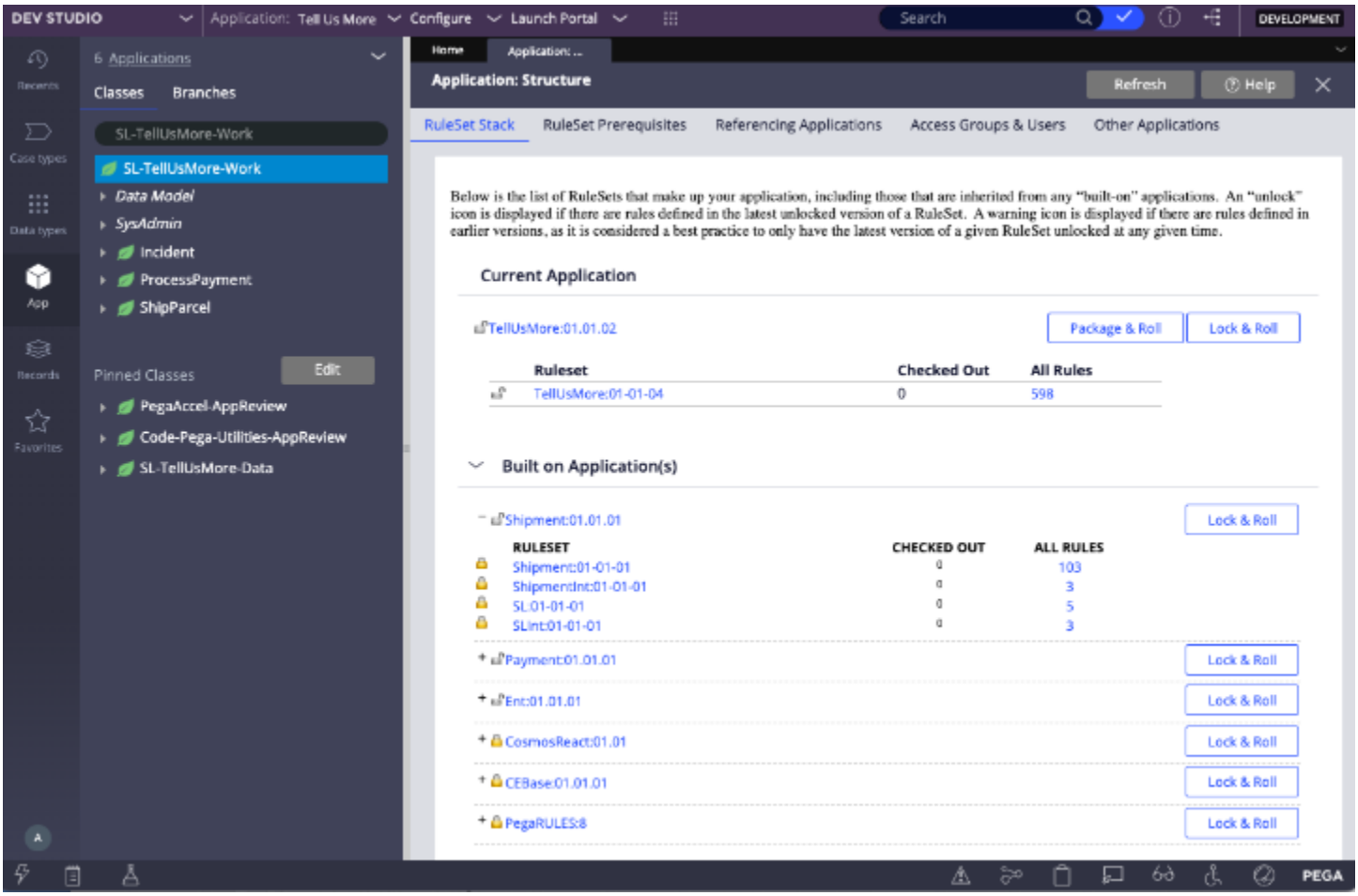
- **Pega Platform layer:** Contains the foundational rules for the Pega Platform.
- **Organization layer:** Contains rules for the entire company or organization.
- **Division layer:** Contains rules for specific divisions within the organization.
- **Framework and Implementation layers:** Contains rules for specific applications.

LOCATION/IMAGE(S): The Enterprise Class Structure is a foundational architectural concept and does not have a single navigation path to a specific artifact. It is defined by the hierarchy of classes within the Pega application. The structure is visible when creating a new class or rule.



Dunnington Software Development Incorporated

Moving At The Pace Of Integrity!



USE CASE/IMPLEMENTATION(S): The ECS is used to ensure that rules are reusable and that applications can be easily extended and maintained. For example, in a banking application, you might define a generic "Customer" class in the Organization layer with rules that apply to all customers, regardless of their specific product. In the Implementation layer, you could have a more specific "Mortgage" class that inherits the generic "Customer" rules but also has specialized rules for mortgage customers. This approach prevents you from having to create the same rule multiple times for different types of customers, saving development time and reducing the risk of errors.

Dunnington Software Development Incorporated

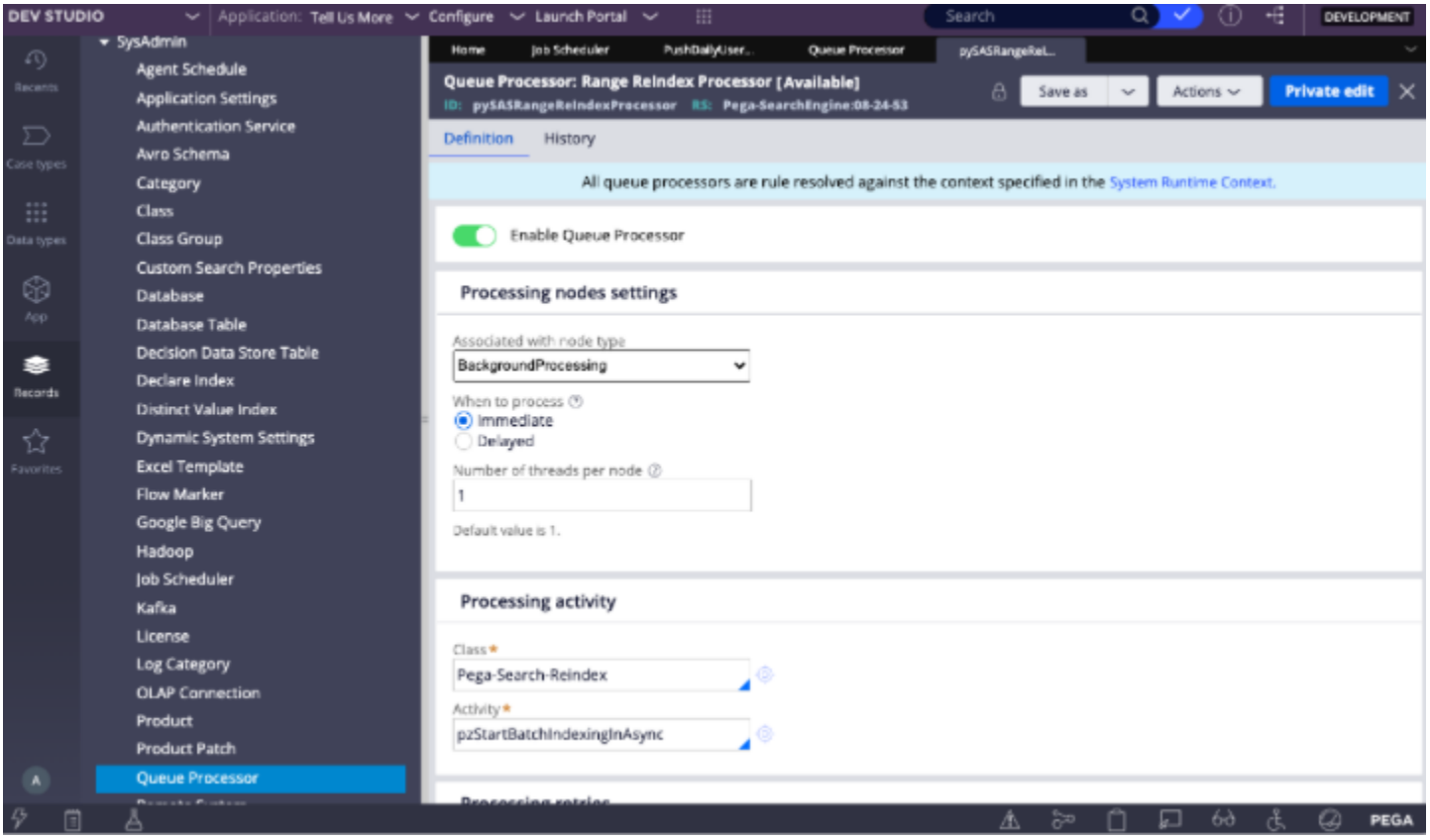
Moving At The Pace Of Integrity!

Question: How is an application created using the New Application wizard?

DEFINITION/DESCRIPTION(S): The **New Application wizard** is a tool that guides developers through the process of creating a new Pega application. It helps to define the application's basic structure, including its name, application type, and initial access groups. The wizard generates the core application artifacts, such as the application rule, initial case types, and a default skin.

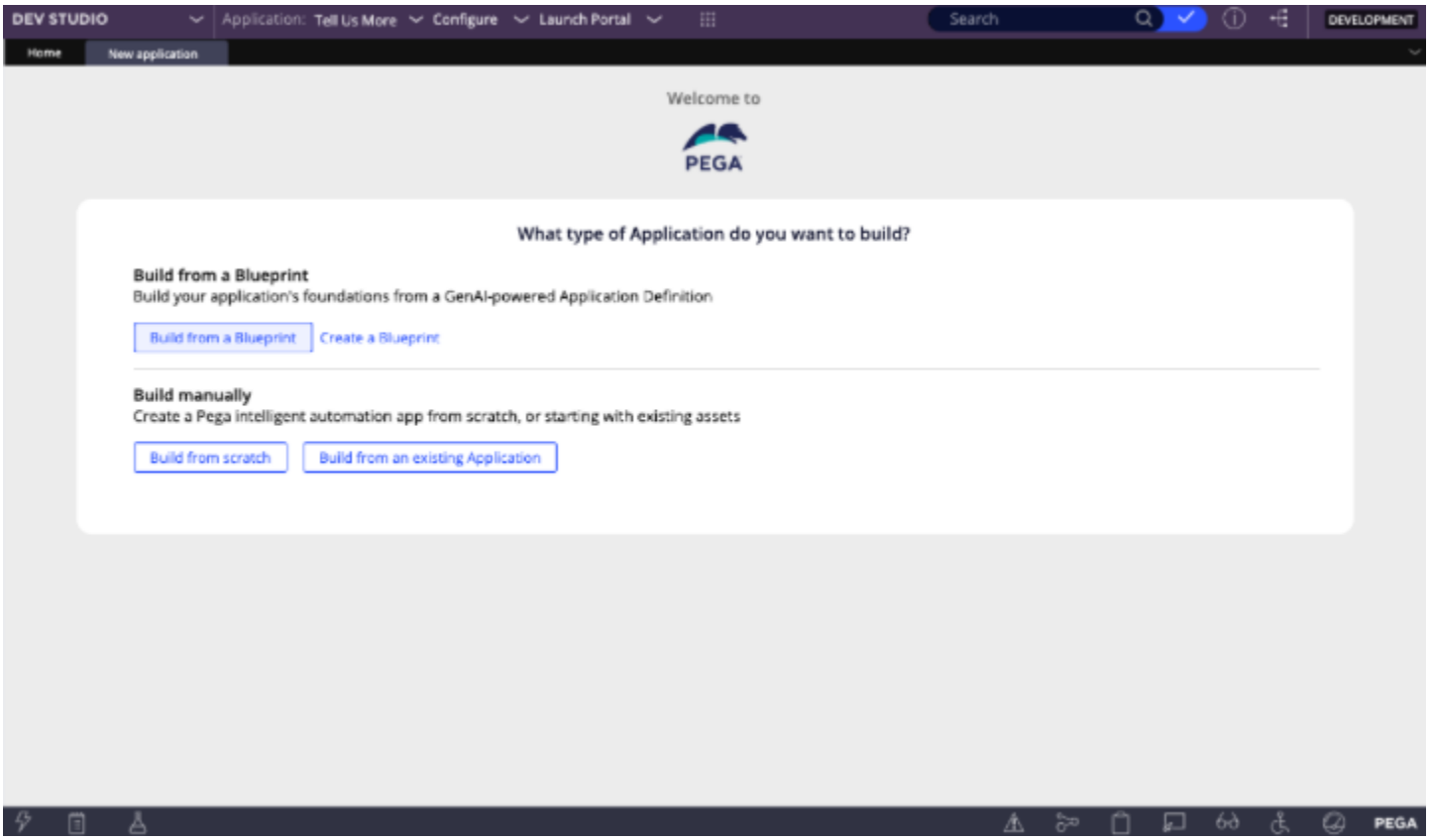
LOCATION/IMAGE(S): To access the New Application wizard from Dev Studio:

- Navigate to **Dev Studio**.
- Click on **New Application**.
- Follow the wizard's steps to name the application, select an application type, and define the initial roles and users.



Dunnington Software Development Incorporated

Moving At The Pace Of Integrity!



USE CASE/IMPLEMENTATION(S): The New Application wizard is used at the beginning of any new application development project to quickly establish the foundational structure of the application. For example, to start a new "Loan Origination" application, a developer would use the wizard to create the application rule and a default "Loan Application" case type. The wizard automatically creates the necessary classes and rulesets, providing a solid starting point for building out the application's business logic.

Dunnington Software Development Incorporated

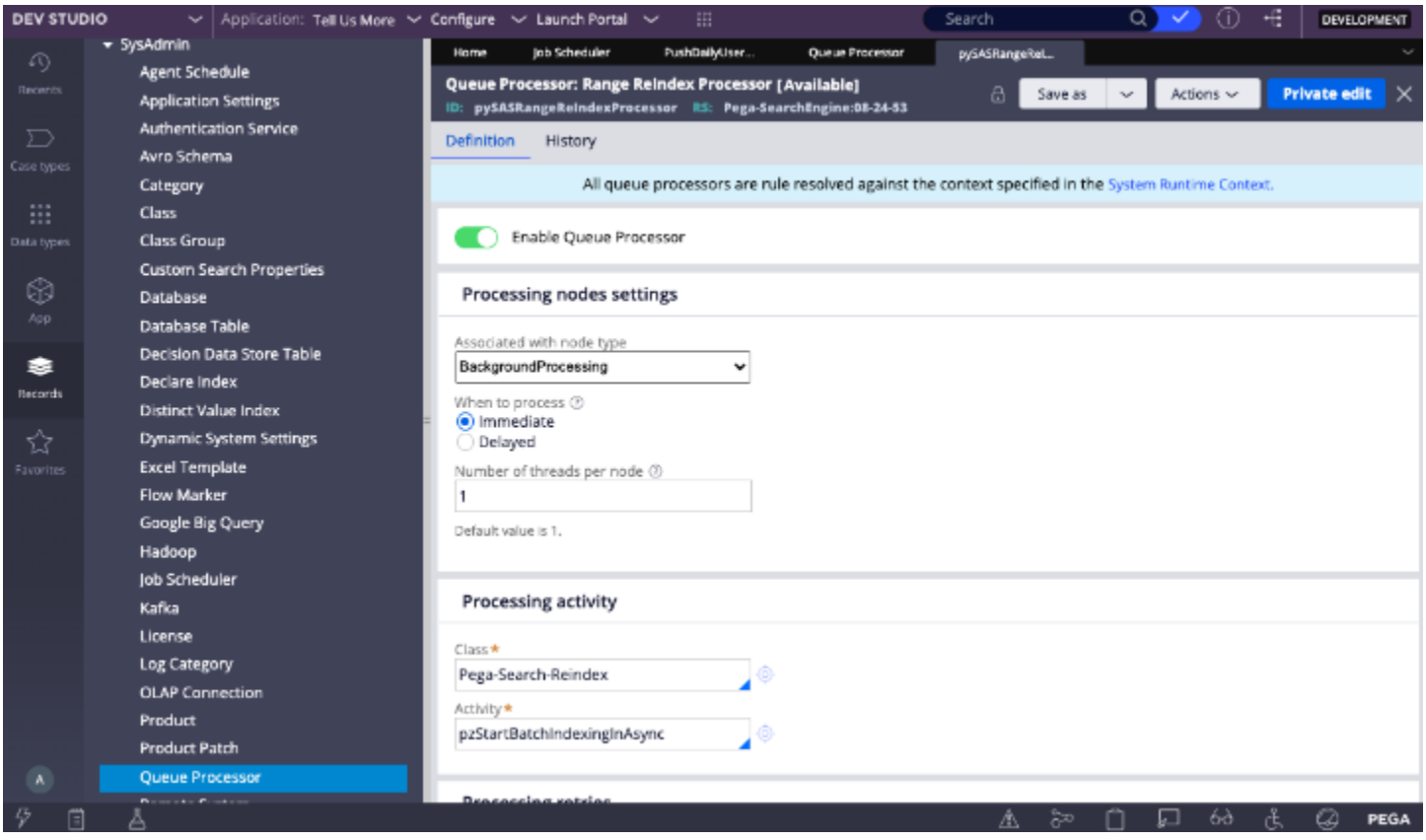
Moving At The Pace Of Integrity!

Question: How does the rule resolution process work, and how can you adjust rule availability?

DEFINITION/DESCRIPTION(S): Rule resolution is the Pega Platform's process for finding the most appropriate rule to execute for a specific context. Pega uses a multi-step algorithm to find the best rule, taking into account factors like the class hierarchy, ruleset list, and rule availability. **Rule availability** is a setting on a rule that determines whether the rule can be used by the system. The availability statuses are **Available**, **Withdrawn**, **Blocked**, and **Final**.

LOCATION/IMAGE(S): To adjust rule availability in Dev Studio:

- Navigate to **Dev Studio**.
- Go to **Records**.
- Open the desired rule (e.g., a Flow Action or an Activity).
- On the rule form, go to the **History** tab.
- In the **Availability** field, select the desired status (e.g., Withdrawn).



USE CASE/IMPLEMENTATION(S): Rule resolution is a fundamental part of how Pega executes business logic. A common use case is when an old version of a rule is no longer needed. By setting the rule's availability to **Withdrawn**, you can ensure that the rule resolution process ignores it and selects a newer, more appropriate rule. This is a crucial practice for managing different versions of rules and ensuring that the correct behavior is always executed.

Dunnington Software Development Incorporated

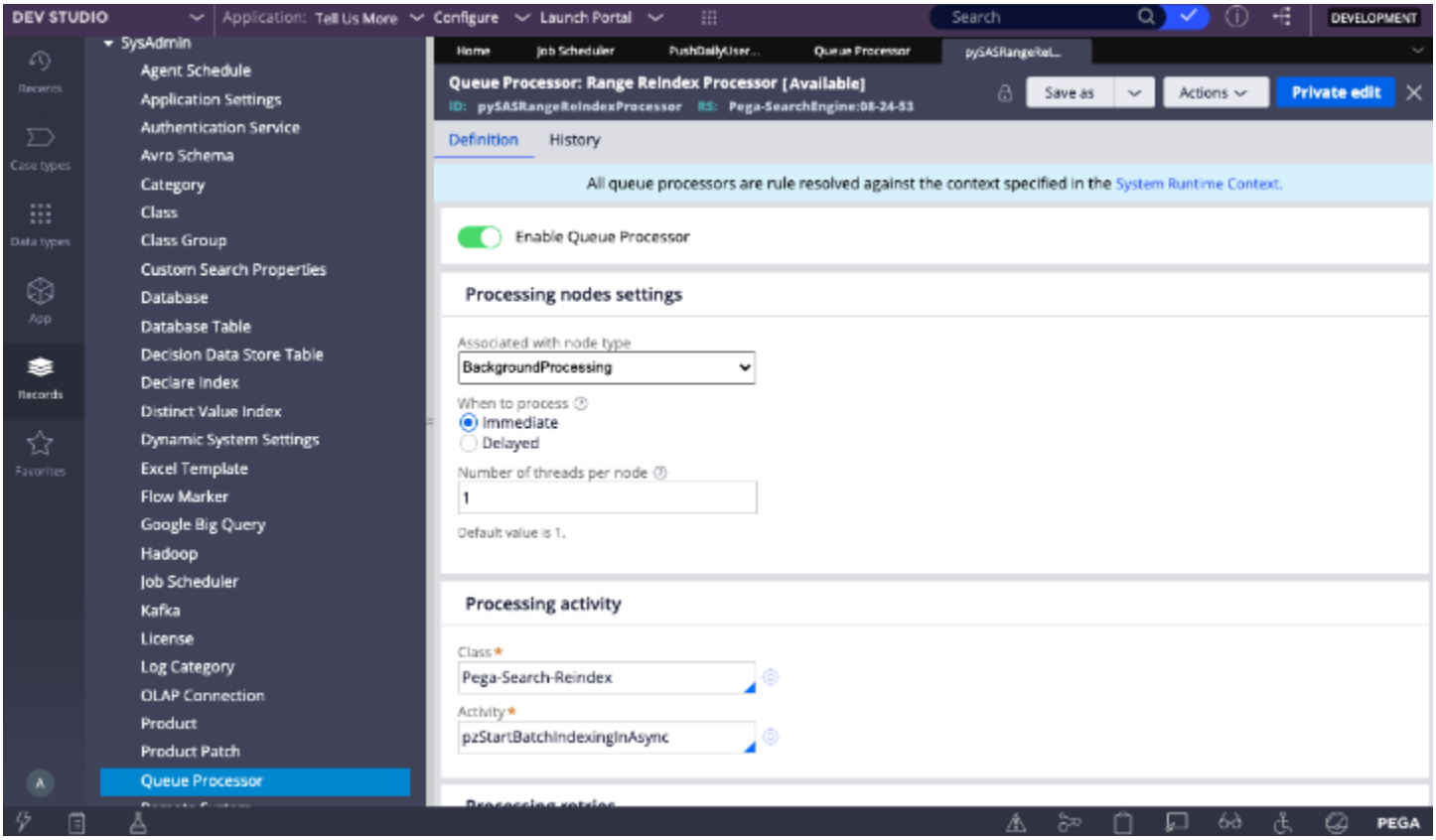
Moving At The Pace Of Integrity!

Question: How does circumstancing affect application behavior?

DEFINITION/DESCRIPTION(S): Circumstancing is a Pega feature that allows you to create a specialized version of a rule that executes under specific conditions. The circumstanced version of the rule is a copy of the base rule with additional conditions, and it is only used when those conditions are met. When the conditions are not met, the base rule is used instead.

LOCATION/IMAGE(S): To circumstance a rule in Dev Studio:

- Navigate to **Dev Studio**.
- Open the base rule you want to circumstance.
- In the toolbar, select **Save As** to create a new version of the rule.
- From the **Save As** dialog, select **Circumstance**.
- Configure the circumstance condition, such as a property value or a date range.



USE CASE/IMPLEMENTATION(S): Circumstancing is used to manage application behavior that changes based on specific conditions without creating multiple separate rules. A common use case is for a shipping application where the shipping fee is different for domestic and international orders. Instead of creating a separate rule for each, you can create one base rule that calculates the domestic fee and then

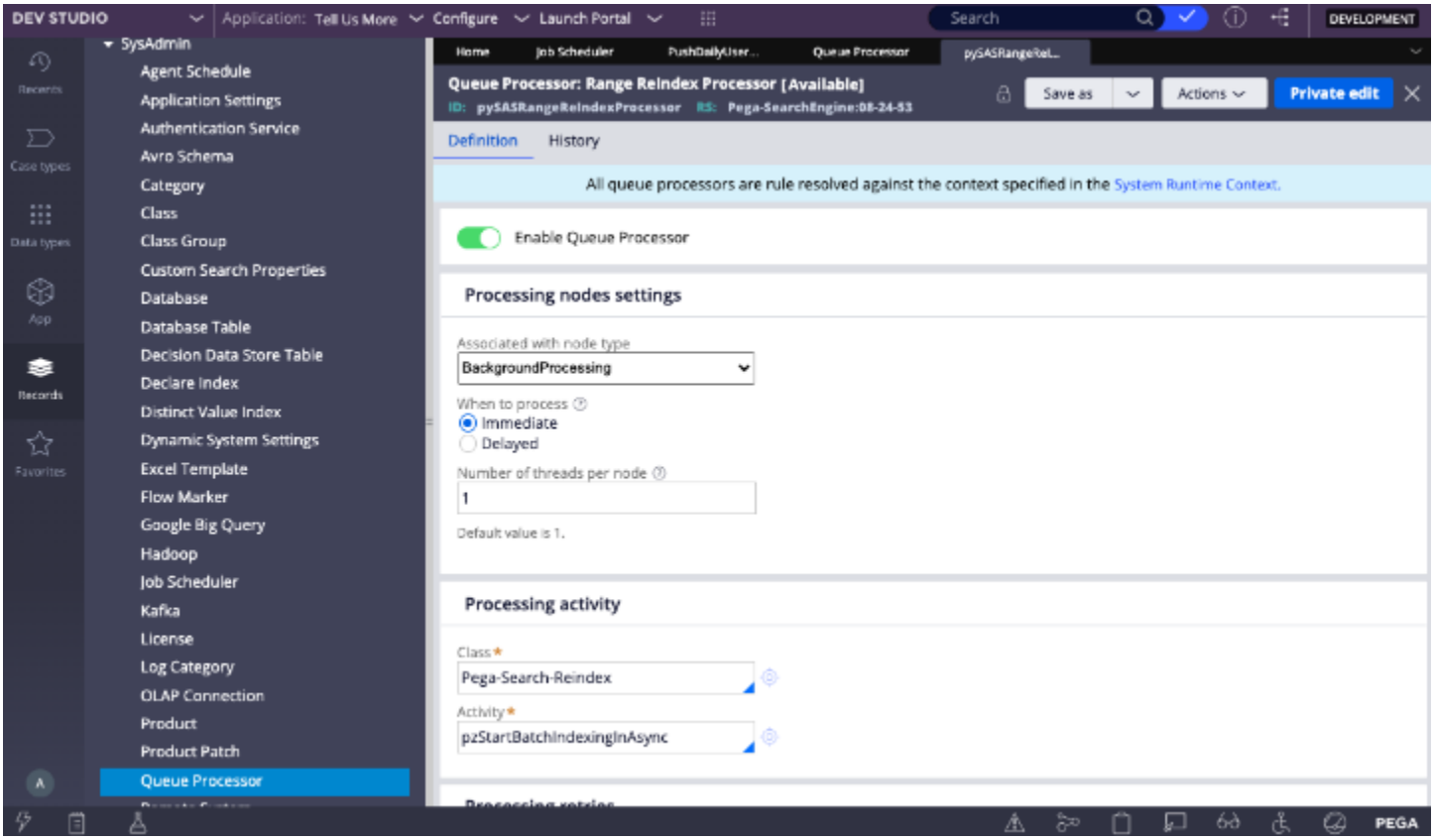
create a circumstanced version that applies when the ShippingCountry property equals "Canada" to calculate the international fee. This approach keeps the business logic organized and easy to maintain.

Question: How are rules circumstanced on a single variable or multiple variables?

DEFINITION/DESCRIPTION(S): Rules can be circumstanced on a single variable using a **single-value circumstance** or on multiple variables using a **multi-variate circumstance**. A single-value circumstance is a simple circumstance based on the value of a single property, such as Country = "USA". A multi-variate circumstance uses a **circumstance template** and a **circumstance definition** to apply a rule based on a combination of property values, such as Country = "USA" AND CustomerType = "Premier".

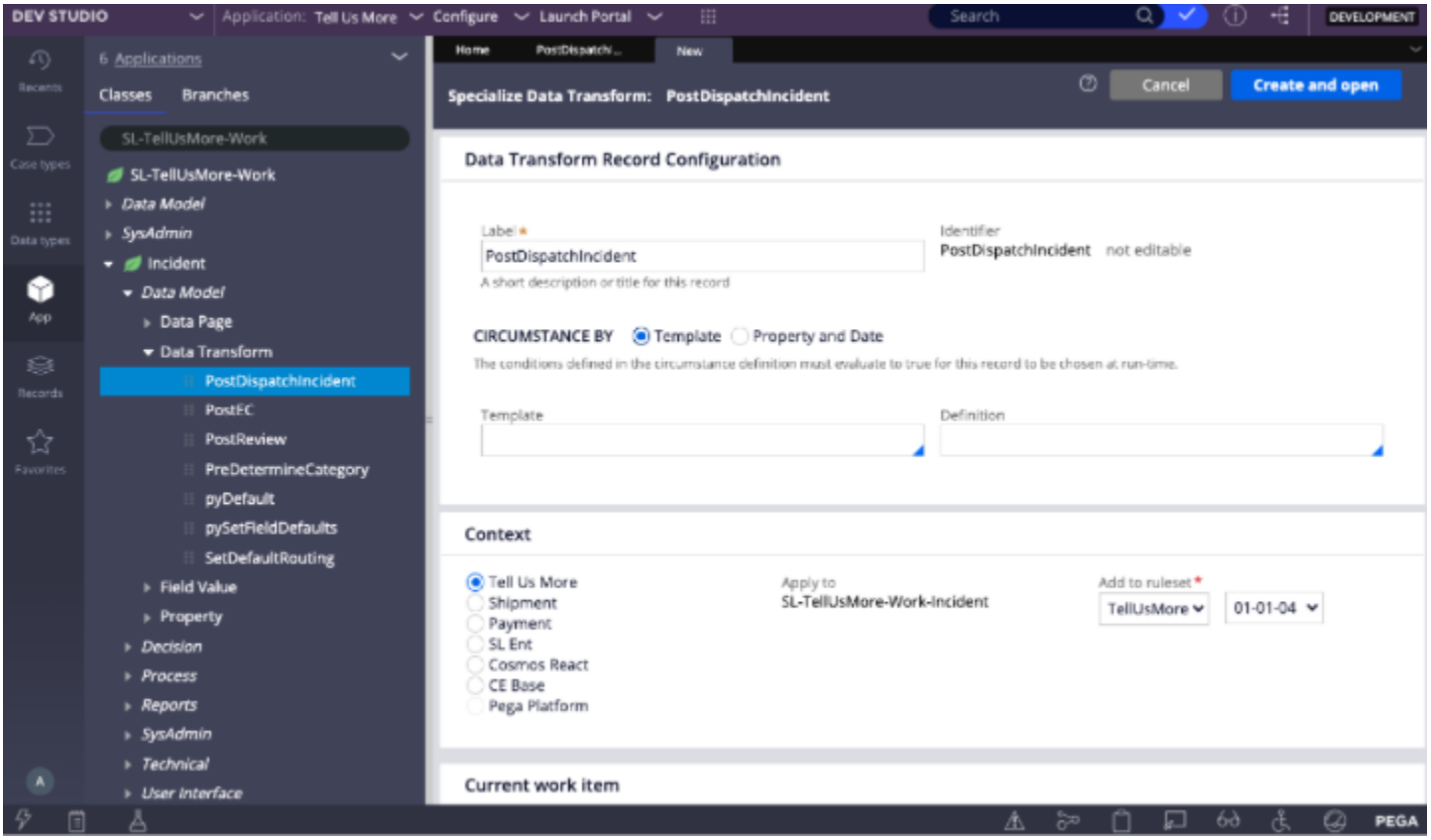
LOCATION/IMAGE(S): To circumstance a rule from Dev Studio:

- Navigate to **Dev Studio**. Click **App**.
- Open the base rule (e.g., property, data transform) you want to circumstance.
- In the toolbar, select **Save As** and then **Specialized by circumstance**.
- To create a single-value circumstance, select **As-is** and specify a single property and its value.
- To create a multi-variate circumstance, select a pre-configured **circumstance template** and provide the corresponding values.



Dunnington Software Development Incorporated

Moving At The Pace Of Integrity!



USE CASE/IMPLEMENTATION(S): A single-value circumstance is ideal for simple business rules that vary based on a single condition. For example, a flow action to "Approve" a loan might have a circumstanced version that is only available if the LoanAmount is less than $10,000. A multi-variate circumstance is used for more complex scenarios. For example, a business rule for a customer's credit limit might vary based on both their CustomerType and their CreditScore. You would create a circumstance template with these two properties and then create a rule for each specific combination of values, such as (CustomerType = "Gold") AND (CreditScore > 750).