

## GUARDRAIL WARNINGS

- While working in your applications you can see some messages. They are known as guardrail warnings.
- **DEFINITION**- Guardrail warnings identify unexpected and possibly unintended situations, practices that are not recommended or variance best practices. It is set of best practices and guidelines that help ensure the quality, performance and maintainability of pega application.
- **Advantages**: Help you develop high quality application, project success, optimal performance, reuse and maintainability for application.
- **How to create a custom guardrail warning**  
Guardrail warnings identify unexpected and possibly unintended situations, practices that are not recommended, or variances from best practices. You can create additional warnings that are specific to the organization's environment or development practices. Unlike rule validation errors, warning messages do not prevent the rule from saving or executing.
- **To add or modify rule warnings**  
To add or modify rule warnings, override the empty activity called @baseclass.CheckForCustomWarnings. This activity is called as part of the Rule-StandardValidate activity, which is called by, for example, Save and Save-As, and is designed to allow you to add custom warnings.  
You typically want to place the CheckForCustomWarnings activity in the class of the rule type to which you want to add the warning. For example, if you want to add a custom guardrail warning to an activity, place CheckForCustomWarnings in the Rule-Obj-Activity class. Place the CheckForCustomWarnings activity in a ruleset available to all developers.
- **Removing or suppressing the default guardrail warnings**  
It is not recommended to remove the out-of-the-box warnings, as they are intended to check if developer's code is complying with good practices. However, in unusual circumstances you may want to do so.  
In such cases, you can suppress the warnings. Create an activity named CheckForCustomWarnings in the Rule class where you want to remove warnings from. When you customize CheckForCustomWarnings, follow the below two steps:  
1)Removal of class specific warnings  
2) Create an activity to remove common warnings across classes which will be included in CheckForCustomWarnings activity.
- **How to configure guardrail warnings in dev studio?**  
Open DEV STUDIO-> configure-> Application-> Quality-> Guardrails-> Compliance score

- **Ten GUARDRAILS to success in Pega.**

1. Adopt an Iterative Project Approach
2. Establish a Robust Foundation
3. Do Nothing That Is Hard
4. Limit Custom JAVA
5. Build for Change™
6. Design Intent-Driven Processes
7. Create Easy-to-Read Flows
8. Monitor Performance Regularly
9. Calculate and Edit Declaratively, Not Procedurally
10. Keep Security Object-Oriented

## Compliance score in guardrail warnings

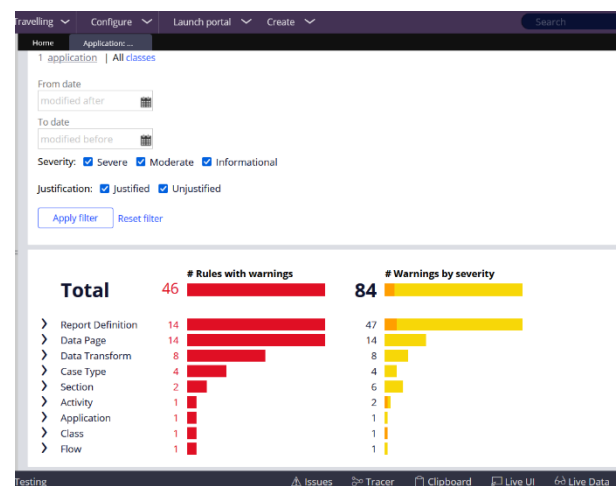
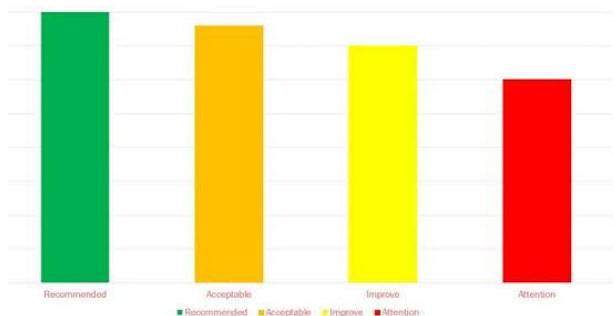
Compliance score is a weighted score that measures overall compliance with pega platform beset practices and guardrails.

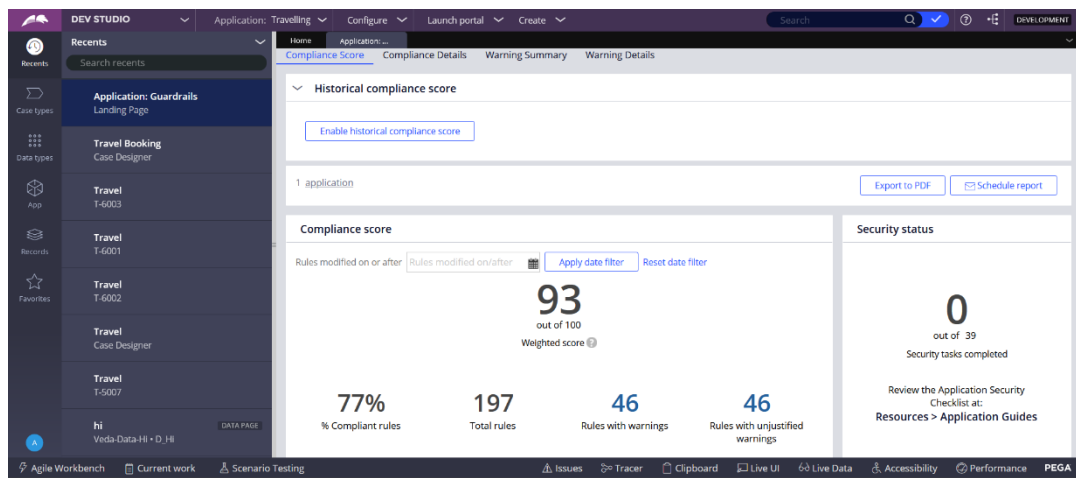
When guardrail warnings decrease compliance score increases

It is calculated based on number of warnings in your application.

- Acceptable when  $\geq 90$ , that means application is in good standing
  - Need to improve when  $80 - 90$ , application needs review for improvement
  - Attention when  $< 80$ , It requires immediate action
- 3 items are generated and displayed in the compliance score of an application
- Rules with warnings in application
  - Rules with unjustified warnings in application
  - Unweighted compliance score

## Guardrails compliance score in Pega





- **How can we improve compliance score?**

1. Reducing the number of warnings in your application
2. Following development best practices
3. Eliminating risks, such as custom code or degraded performance
4. Reconfiguring rules as necessary
5. Analysing your compliance data to find the root cause of a low compliance score
6. Creating unit test cases