**Any Pega Rule we create in Application it is saved in Rule base under some Instances ( PR4 Tables)**

**Data Tables are stored under PR Tables**

**==========**

**Rule Criteria's:**

**1- Circumstance**

**2- Class & Ruleset Hierarchy**

**3- Rule Authorization Process/Access Group**

**4- Rule Cache**

**5- Rule Availability**

**Ruleset hierarchy:**

**User Defined Ruleset - User Defined Class - 1st Pattern**

**Predefined Ruleset - Predefined Classes - 2nd- Direct**

**Class Hierarchy:**

**To search for any rule it will follow below class hierarchy.**

**1st - Current Class/Unit**

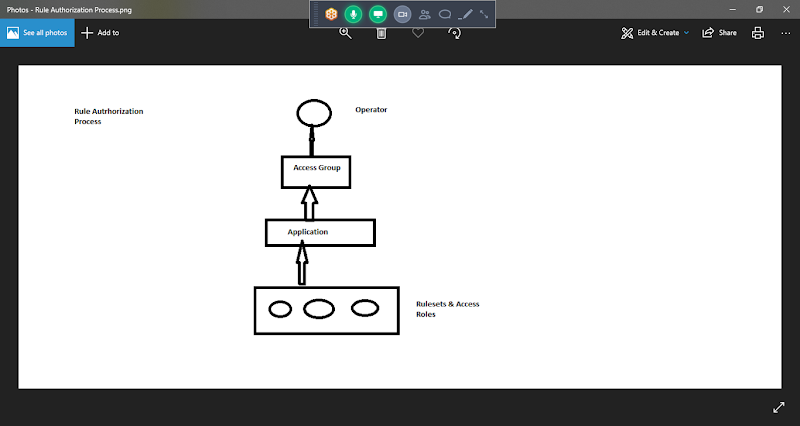
**2nd- Pattern ( Division & Organization )**

**3rd- Direct  (Work-Cover-/ Work-)**

**4th - @baseclass**

**Rule Authorization Process:**

**we can access the rule from our application within same access group.**



**Access Group: Is a interface b/w Operator & his application , Portals & Access Roles**

**Access Group Contains 3 things:**

**Application Access ( Application Name)**

**Portal Access( Developer, Manager & User Portals)**

**Access Roles & Privileges**

**Rule Cache:  Cache Memory**

**If a Rule is Available in cache, it will take from Rule cache else it will take from the Rule base Table**

**If we Login to Application & Executes Customer Flow it will Pick All the Rules used in Customer Flow from Rule Base Table.**

**If we Executes same flow again it will access the Rules from Rule Cache.**

**Rule Availability:**

**Availability Types: we have 5 types**

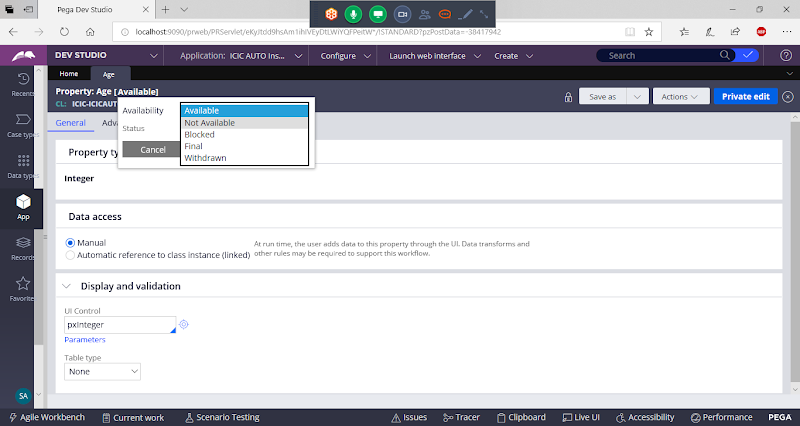
**Available ( Default)**

**Not Available**

**Final**

**Withdrawn**

**Blocked**



**Available for Processing:**

**Available: If Rule is available means it is available to update, it will display the rule in Latest Version from Current Class. Copy from One Ruleset Version to other Ruleset Version ( Can Modify), if we have same rule in multiple version then it will automatically call the rule from latest version.**

**Final:  Can't Copy from Ruleset Version to Other Ruleset Version( Can't Update). Final always we have to set for Highest Version only**

**Ignored in Processing:**

**Not Available: Current Version it will Ignore**

**Age: 01-01-01, 01-01-02( Not available), then it will access age from 01-01-01**

**We have to set NotAvailable only for highest version only**

**Blocked : Any where the rule is blocked, preventing usage of the rule in PRPC is Blocked**

**Age: 01-01-01, 01-01-02( Blocked),   Every where rule will be Blocked**

**Withdrawn: If the rule withdrawn, then all the versions in the current class( Class group)  will be ignored & proceeds in the Inheritance hierarchy of Division, then Organization,  Work-Cover-, Work-, @baseclass**

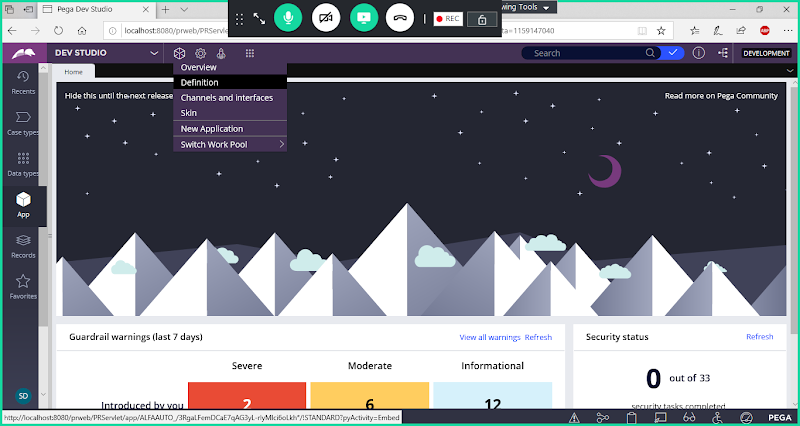
**Work Class: Age: 01-01-01, 01-01-02( Withdrawn) - All versions in Current Class will be WIthdrawn**

**Division class: Age: 01-01-02**

**Organization Class: Age 01-01-02**

**Availability Types:**

**Open Application Definition:**

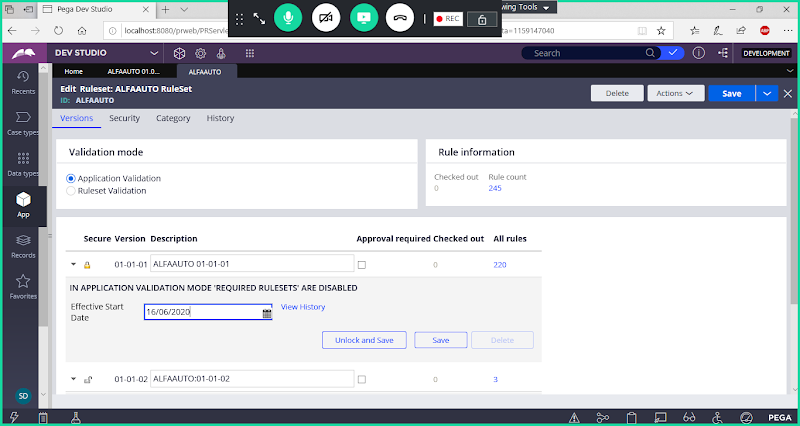


**Open Ruleset:**

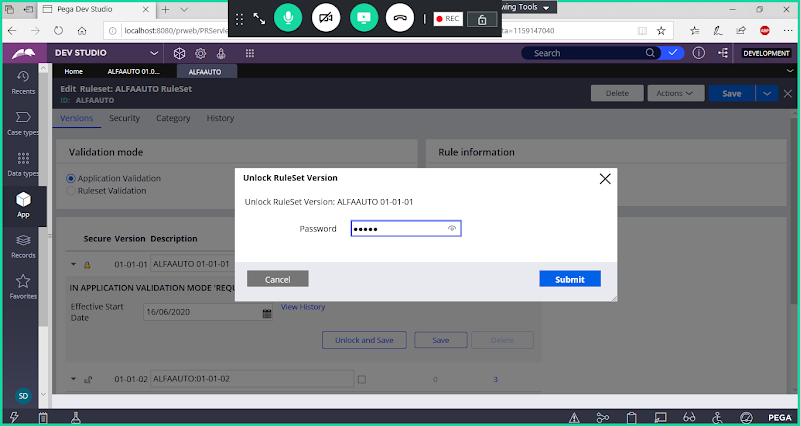
**01-01-01**

**01-01-02**

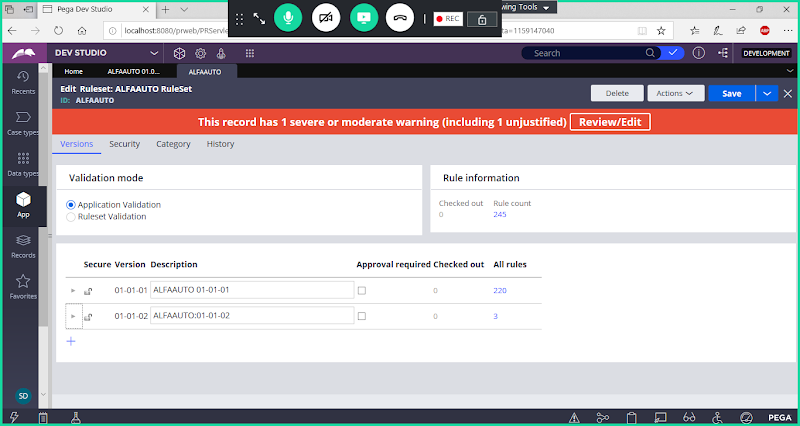
**Release the Lock for Old Version ( 01-01-01)**



**Click on Unlock & Save**



**Click on Submit**



**Save Ruleset Form**

**Understanding Availability Types:**

**Unlock the Ruleset Versions (  01-01-01, 01-01-02)**

**Add Property: A1 ( Radio Button) - Local List - Add Value**

**V1: Work Class group**

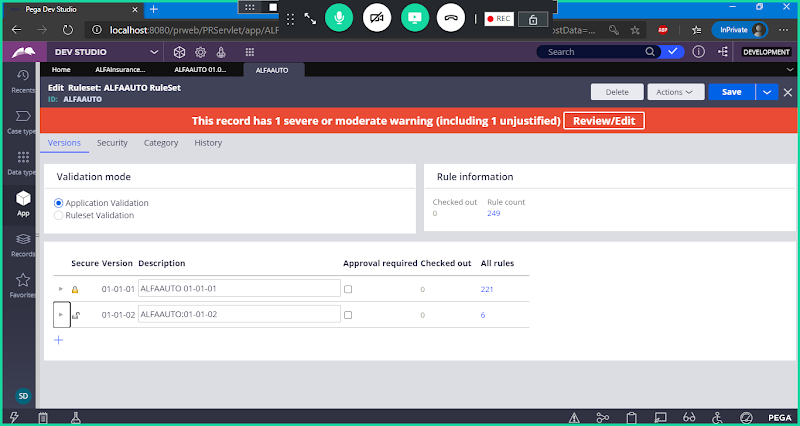
**V2: Work Class Group**

**V2: Division Class**

**Similarly create properties in other classes as mentioned above.**

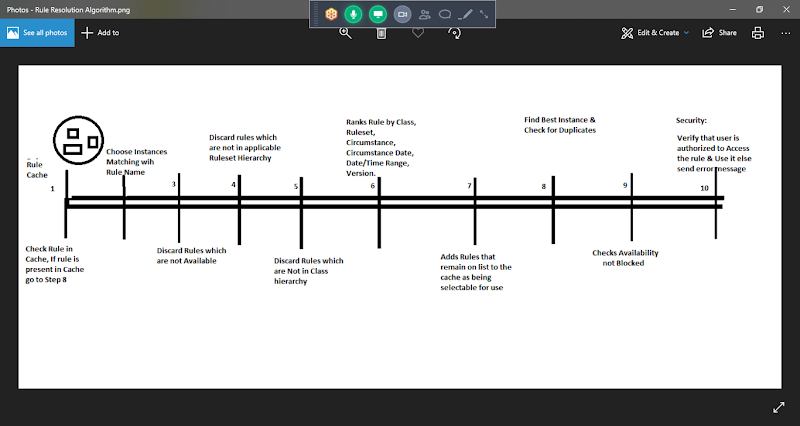
**==============**

**After Verifying all the Availability Types, Lock Ruleset Version ( 01-01-01)**



**Rule Resolution Algorithm:**

**All the above criteria's we combine into single Algorithm called as Rule Resolution Algorithm.**



**Scenario1: If Rule Available in Rule Cache**

**Step1-> Step 8 -> Step 9 -> Step10**

**Scenario2: If Rule is not available in Rule Cache ( It will Pick the Rule from the Rulebase Table)**

**Step2-> Step 3-> Step4-> Step5-> Step6 -> Step7 -> Step 1( Rule Cache)**

**Step1-> Step 8 -> Step 9 -> Step10**

**=====================**