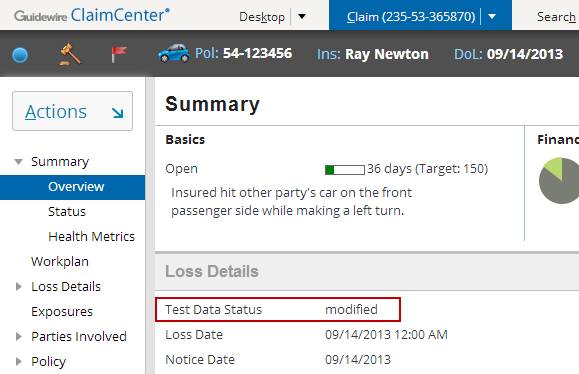
The ClaimCenter Data Model

**NOTE**: A development instance of ClaimCenter is required for this and all the labs in this course.

Exercise 1: Configuration

Configure ClaimCenter to meet the following customer requirement from Acme Insurance.

During the implementation of ClaimCenter, the team for Acme Insurance will be working with imported test data. When viewing any claim, developers want to know if the claim was created by a user or is imported test data. If the claim is from imported test data, then they want to know if it has been modified since the data was imported.



1. Add a single field to the Claim entity named **TestDataStatus\_Ext** according to the following:

* This field can contain a string up to 15 characters in length.
* It is **ok** for the field to have null values.
* The field must have a default value of “unconfirmed”.

1. Add a field to the **Claim Summary** detail view, in the **Loss Details**. The field should have the following attributes:

* The field appears above the **Loss Date** field. However, it appears only if the claim is a test data claim.
* The field maps to **TestDataStatus\_Ext**.
* The field's label is Test Data Status.
* The field is not editable.

1. Add a rule to the Claim Preupdate rule set which meets the following requirement:

* The rule sets the value of **TestDataStatus\_Ext** to modified if this is a test data claim.

1. The ClaimCenter implementation team has provided the following additional information:

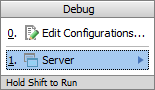
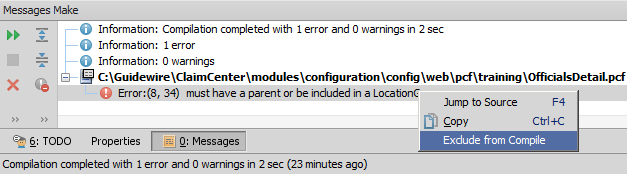
* Most of the test data has a public ID which starts with the characters demo\_sample. Therefore, to determine if a claim is a test data claim, check to see if its public ID starts with demo\_sample.

1. You can use the startsWith string method, which has the following syntax:  
   gw.api.util.StringUtil.startsWith( stringToCheck, stringToCheckFor )

The logic needed to determine whether the **TestDataStatus\_Ext** field is visible or the Claim preupdate rule described below should execute are identical. For this reason, both the rule and the visible field on **TestDataStatus\_Ext** should reference an enhancement. That function should return true if the data is demo data and false otherwise.

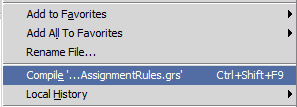
NOTE: When testing your gosu enhancements for this lab, a server restart is required. You may perform all the configurations needed for this lab and restart the server (since you have made changes the data model).

*In ClaimCenter 8.x and later, changes to gosu enhancements require a server restart to take effect. Changes to rule sets (not PCFs) can be reloaded using “Make Project” or “Reload Changed Classes” in Guidewire Studio*. *Any other changes, such as data model changes, typelist modifications, or changes to files such as config.xml still require a server restart. ALT+SHIFT+L is not the recommended approach and you no longer are presented with a “banner” message in the browser alerting you that changes have been made. You may continue to use ALT+SHIFT+L during training to view changes to PCFs/web templates only.*How to run Make Project/Reload Changed Classes in Studio 8.x:

1. With the server **stopped**, run the server in a debug process (instead of in a separate console). In Studio, choose “Run > Debug”.
2. Chose “Server”:   
   
3. ClaimCenter will start in Studio (Console tab). Verify that you see the **\*\*\* ClaimCenter ready\*\*\*** message. If, during your work, one or more files have errors (such as work you are putting on hold), you may exclude the file from compile, **and then re-run step #2 again:**  
   
4. To propagate your changes to rule sets, choose **“Run > Reload changed classes”** or press **CTRL+F9** to make project. Alternatively, you may click the **“Make Project”** button:  
   

**Summary of 8.x reloading/testing process:**

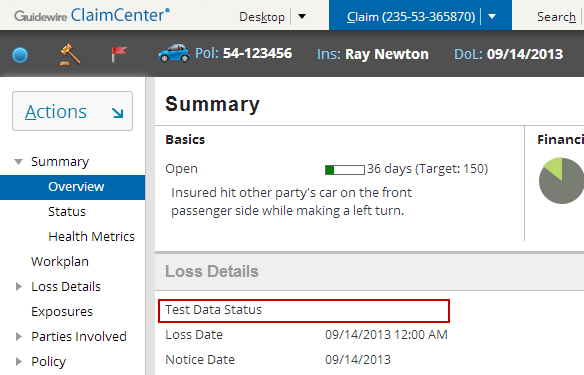
|  |  |
| --- | --- |
| **Change** | **How to reload** |
| Typelist, Data Model, Config Files, Gosu Enhancement | Restart server |
| PCF (UI) | ALT+SHIFT+L |
| Rule Set | Make Project/Reload Changed Classes |

1. If you run into difficulty with reloading rules, or wish to *include* a file back into compile, try right-clicking on the file itself, and running a compile:  
   

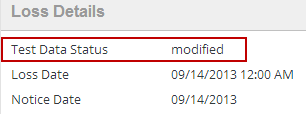
Test Case

When you have completed your configuration, run the following test case:

1. Navigate to the Ray Newton claim **235-53-365870** from the sample data. (This test will work for many of the sample data claims, but not all of the sample data claims have publicID’s that begin with “demo\_sample”, so use this claim to ensure an accurate test.) Verify that the "Test Data Status" field appears. The field should be blank.



1. Modify the claim and verify that the field changes to "modified".



1. Create a new claim using the quick claim wizard. Once the claim is complete, verify that the "Test Data Status" field does not appear.

