Transaction Validation Rules

Exercise 1: Investigation

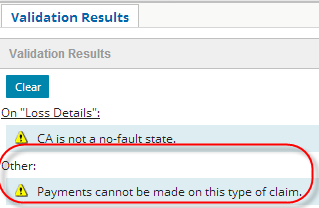
Create a transaction set validation rule named "Observation Rule". Cut and paste the following code into the CONDITION. Then, answer the question below.

return transactionSet.Claim.LossType==LossType.TC\_PR

1. Under what circumstances will this rule execute? Write your explanation as if you were discussing the issue with a non-technical claims adjuster.
2. Cut and paste the following incomplete line of code into the ACTION. Then, complete the code so that it displays a warning with the text message "Payments cannot be made on this type of claim." What is the complete line of code?

 transactionSet.reject( /\* fill in the parameters \*/ )

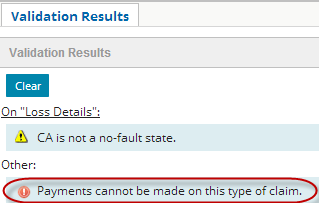
1. Reload changed classes in Studio (Run > Reload Changed Classes). Test your answer by logging on to ClaimCenter as Ronald Barnes (rbarnes) and creating a new reserve line on any claim he has access to.



NOTE: Depending on the claim you use, you may or may not see a warning about no-fault states.

NOTE: You may need to restart the server if reloading changed classes does not reload any classes.

1. The validation rule is a warning about payments. Why does it display when you create a reserve line?  
   Can you add the reserve anyway? Go ahead and complete the reserve transaction.
2. Now, modify the code in the Rule CONDITION so that the rule action executes only if the transactionSet contains payment transactions. Also modify the line of code in the Rules ACTION so that it displays an error with the same text message. What is the correct code?
3. Reload changed classes. Test your answer by creating a new reserve line on a Ronald Barnes claim, and then by creating a payment against the reserve created above. What happens?



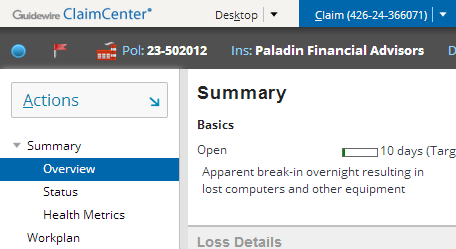
1. Disable the “Observation Rule”.

**Exercise 2: Configuration**

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

**Requirement 1: Preventing Indemnity Payments on Flagged Claims**

As you may recall, claim exception rules are run periodically to identify claims that "require attention". When a claim requires attention, it is usually flagged. This creates a visual cue in the info bar for the supervisor, who is responsible for reviewing the claim, attending to the issue, and then removing the flag.



Acme Insurance does not want new payments to be made on a flagged claim, as the issue creating the flag could alter the payment or even make it unnecessary.

1. Create the rule(s) needed to enforce these behaviors:

* A user cannot make an Indemnity payment on a flagged claim.

1. The ClaimCenter implementation team has provided additional notes:

* The error message should read "You cannot create indemnity payments on flagged claims."
* The requirements document does not specify how to identify that a claim is flagged. However, you should be able to determine this by reviewing the data dictionary.
* The rule should only prevent **new** payments and allow changes to existing payments (such as voiding a check)

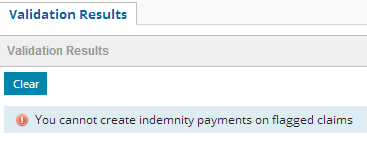
**Test Cases: Requirement 1**

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

1. Remember to reload your configuration (reload changed classes) before testing. You may need to restart the server if reloading changed classes does not reload any classes.
2. Log on to ClaimCenter as Andy Applegate. Navigate to claim # 235-53-365870, a flagged claim, and ensure it is assigned to Andy Applegate (assign it to him if necessary.) If the claim and at least one of its exposures is not at Ability to Pay, then bring the claim and at least one of its exposures to Ability to Pay.

If claim # 235-53-365870 is not flagged or suitable for use, you can select another flagged claim. Confirm that the claim you select is not already under financial holds, since that will restrict the creation of indemnity payments.

1. The three situations that would place a claim under financial holds are:  
   - Coverage is in question  
   - Claim is marked incident-only  
   - Policy is unverified
2. Attempt to create an indemnity payment on the claim. This should fail.



1. Restart the Check Wizard and change to create an **expense** payment on the claim. This should succeed without errors. You may receive a warning but this can be ignored.
2. Stop payment on check **# 10436** (for Ray Newton). For instructions on how to stop payment, refer to the last page of this lab. Stopping payment follows a very similar process to voiding a check.   
     
   Stopping payment should succeed without errors because your rule only prevents the creation of payments and allows changes/edits to existing payments. The status should change to “Pending stop”. (If there is no such check, or it has already been stopped, test stop payment on a check on another flagged claim).

Requirement 2: Prevent the Voiding of Issued Checks

Acme Insurance issues payments through the Millennium Mutual bank. The agreement between the two companies is that once a check has been issued a check number, it cannot be voided. At this point, Acme Insurance must request a stop payment on the check.

12. Create the rule(s) needed to enforce these behaviors:

* A user cannot void a check with a check number.
* The Check Number field should be visually tagged by the error

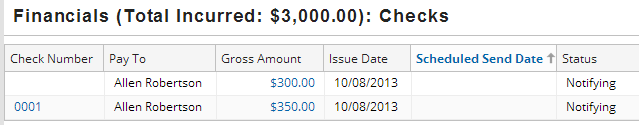
13. The ClaimCenter implementation team has provided additional notes:

* To determine if a user is attempting to void a check, you should check to see if the check's status is being changed. However, keep in mind that the "Voided" status is determined by the external check processing system. To prevent the voiding of a check within ClaimCenter, you must look for a status which is assigned by ClaimCenter.

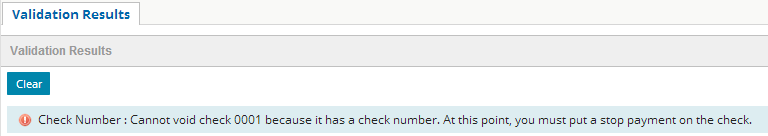
Test Cases: Requirement 2

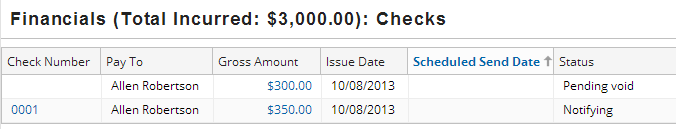
When you have completed your configuration, run the following test case:

14. Log on to ClaimCenter as Andy Applegate. Using any claim on which payments can be made, create two **manual** checks. (See the next page for an overview of how to create manual checks.) Make sure that the first check has no check number and the second check has a check number.



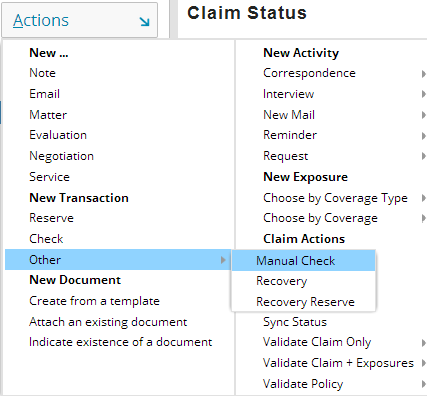
15. After the checks has been created, attempt to void both checks. (See the next page for an overview of how to void checks.) Verify that the second check with a **check number** **cannot be voided.**



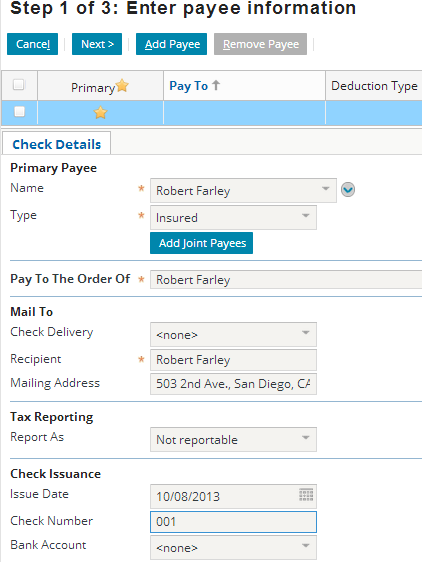


How to Create a Manual Check with a Check Number

1. Select New Transaction > Other > Manual Check.

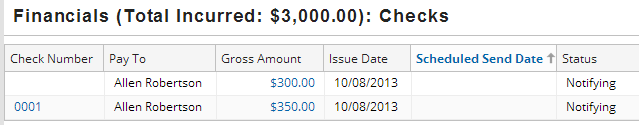


2. In step 1, specify a check number. Otherwise, complete the wizard as if it was a regular check.

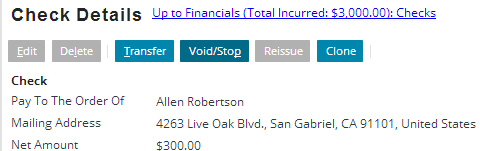


**How To Void a Check**

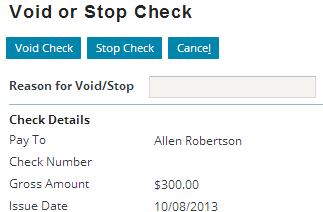
1. From the Checks lists, click the amount of the check.



2. In the Check Details screen, click "Void/Stop".



3. In the Void or Stop Check screen, click "Void Check" and confirm the dialog window as well.



4. If the voiding is successful, the Status changes to "Pending void".

