ClaimCenter 10 Configuration: Kickstart

Labs and Tutorials

Document Published: 2020-11-11

Table of Contents

Introduction

Welcome to the Guidewire **ClaimCenter 10 Configuration: Kickstart** course!

The Student Workbook you will lead you through various course module labs and provide additional information. The module numbers correspond to the module numbers in your training. Complete the assigned labs to the best of your ability.

Lesson 1

Configuring the ClaimCenter User Interface

As a developer, you must configure ClaimCenter to work the way your users do.

Pages and Location Groups determine the way the information users see is organized. Understanding how to configure them will allow you to organize information to suit your user's day-to-day needs.

1.1 Adding a view to a page

Succeed Insurance has specified that the Claim Summary – Overview should contain a list view showing the Name, Type, and Status of any documents associated with the Claim. In this lab, you will implement this business requirement. This list should be displayed at the bottom of the Summary and should appear only if there is at least one document associated with the Claim.

Prerequisites

For this lab, you will use ClaimCenter, Guidewire Studio, and a supported web browser.

ClaimCenter Server should be running in debug mode when you start the lab.

The default URL for ClaimCenter is http://localhost:8080/cc/ClaimCenter.do.

1.1.1 Lab

Activity

Tip

The easiest way to open a file (with a known name) in Guidewire Studio is to type **Ctrl+N** and begin typing the name of the file. When the name of the file appears in the dropdown, click it.

1. **Determine what existing PCF files will be used to implement this business requirement..**

   a) Log in to ClaimCenter as **aapplegate/gw**.
   b) Navigate to the **Ray Newton** claim (235-53-365870)
   c) In the Claim Summary Overview, type **Alt+Shift+E** to open the current location in Studio.

d) What is the name of the Page?

2. Create the List View needed for this business requirement.

a) In ClaimCenter Studio, open the List View **DocumentsLV.pcf**.

b) In the Project pane, navigate to **config ➔ web ➔ pcf ➔ claim ➔ summary**, right-click the Summary folder, and select **New ➔ PCF File**

c) In the PCF File dialog, select **ListView** and enter the name **ClaimSummaryDocs_Ext**.
The editor will attach the suffix LV to the PCF File name.

d) Give the List View one Required Variable, named **Claim** and of type **Claim**.

e) Add a **Row Iterator** element to the List View.

f) Add a **Row** element to the Row Iterator.

g) Set the properties of the Row Iterator as follows:

editable	**false**
elementName	**document**
value	**Claim.Documents**

NOTE: Recall that you can create a new Display Key in the label field using Alt + Enter

h) Add a **Text Cell** to the Row.

editable	**false**
id	**Name**
label	**DisplayKey.get("LV.Claim.Documents.Document.Name")**
value	**document.Name**

i) Add two more **Text Cells** to the Row:

editable	**false**
id	**Type**
label	**DisplayKey.get("LV.Claim.Documents.Document.Type")**
value	**document.Type**

editable	**false**
id	**Status**
label	**DisplayKey.get("LV.Claim.Documents.Document.Status")**
value	**document.Status**

3. Add the List View you created to the Claim Summary Page.

a) Navigate to the **ClaimSummary** Page.

b) At the bottom of the page, add a Panel Ref.

c) Set its properties as follows:

def	**ClaimSummaryDocs_ExtLV(Claim)**
id	**ClaimDocs**
visible	**Claim.Documents.length > 0**

4. Verify your work.

a) Switch to a browser open to ClaimCenter.

b) Type **Alt+Shif+L** to reload the PCF files.

c) If necessary, log in as **aapplegate/gw** and navigate to the **Ray Newton** claim.

d) Scroll to the bottom and verify that the Documents list appears as expected.

e) Navigate to the **Bill Kinman** claim and verify that the Documents list does not appear.

STOP

1.2 Adding a page to a location group

Succeed Insurance has specified that, on the Claim menu, the Litigation menu link should be included in the Plan of Action group, at the bottom of the group.

1.2.1 Lab

Activity

1. **Determine the PCF elements involved.**

   a) In Guidewire Studio, navigate to the **Claim** Location Group.
   b) Right-click the **Litigation** Location Ref and select **Cut**.
   c) Find the Plan of Action group, note its name, and navigate to it.
   d) Right-click and select **Paste**.
   e) Paste the Litigation location ref at the bottom of the location group.
   f) Delete the location ref's **icon** property.

2. **Verify your work.**

   a) In ClaimCenter, reload the PCF files.
   b) Navigate to the Summary Page of any Claim.
   c) Verify that the Litigation menu link appears ***only*** at the bottom of the Plan of Action group.

STOP

1.3 Solutions

1.3.1 Adding a view to a page

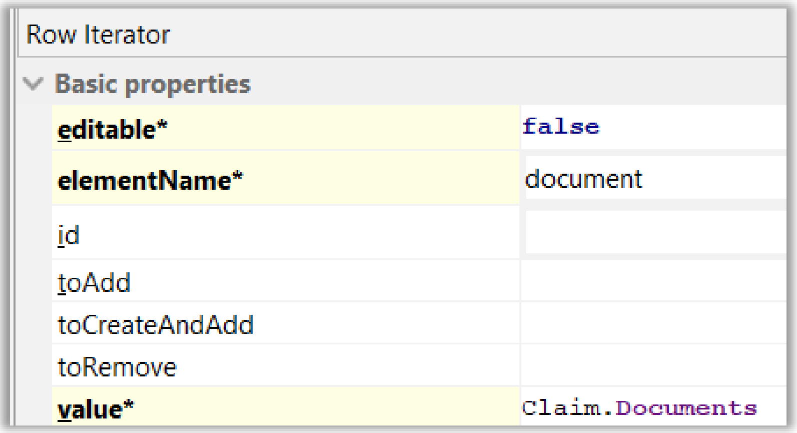
Solution

Exact details on how to complete the lab.

1. **Determine what existing PCF files will be used to implement this business requirement..**
   a) Log in to ClaimCenter as **aapplegate/gw.**
   b) Navigate to the **Ray Newton** claim (235-53-365870)
   c) In the Claim Summary Overview, **type Alt+Shift+E** to open the current location in Studio.
   d) What is the name of the Page?
   ClaimSummary

2. **Create the List View needed for this business requirement.**
   a) In ClaimCenter Studio, open the List View **DocumentsLV.pcf**.
   b) In the Project pane, navigate to **config ➔ Page Configuration ➔ pcf ➔ claim ➔ summary**, right-click the Summary folder, and select **New ➔ PCF File**
   c) In the PCF File dialog, select **ListView** and enter the name **ClaimSummaryDocs_ExtLV**.
   d) Give the List View one Required Variable, named **Claim** and of type **Claim**.
   e) Add a **Row Iterator** element to the List View.
   f) Add a **Row** element to the Row Iterator.
   g) Set the properties of the Row Iterator as follows:

editable	**false**
elementName	**document**
value	**Claim.Documents**



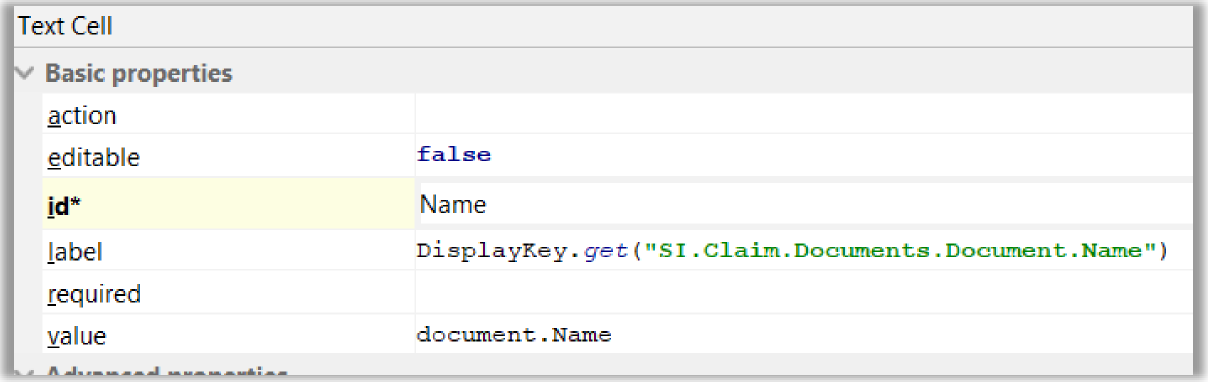
Row Iterator

Basic properties

editable*	false
elementName*	document
id	
toAdd	
toCreateAndAdd	
toRemove	
value*	Claim.Documents

h) Add a **Text Cell** to the Row.
NOTE: Recall that you can create a new Display Key in the label field using Alt + Enter

editable	**false**
id	**Name**
label	**DisplayKey.get("SI.Claim.Documents.Document.Name")**
value	**document.Name**



Text Cell

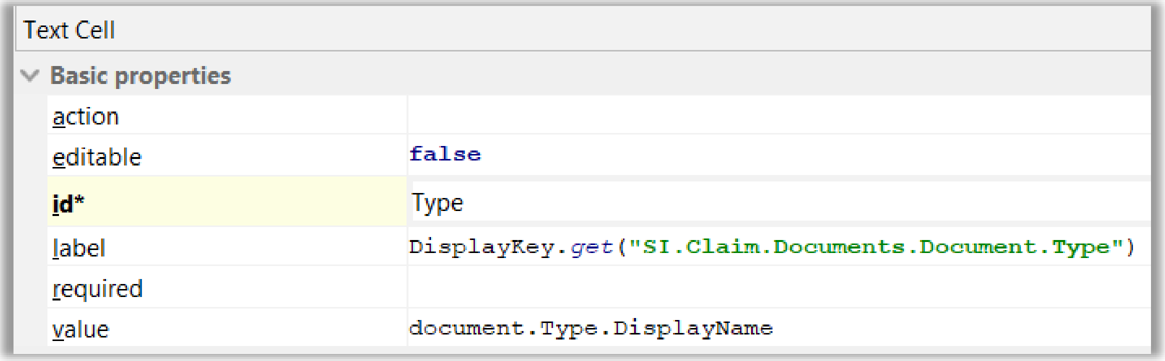
Basic properties

action	
editable	false
id*	Name
label	`DisplayKey.get("SI.Claim.Documents.Document.Name")`
required	
value	`document.Name`

Advanced properties

i) Add two more **Text Cells** to the Row:

editable	**False**
id	**Type**
label	**DisplayKey.get("SI.Claim.Documents.Document.Type")**
value	**document.Type.DisplayName**

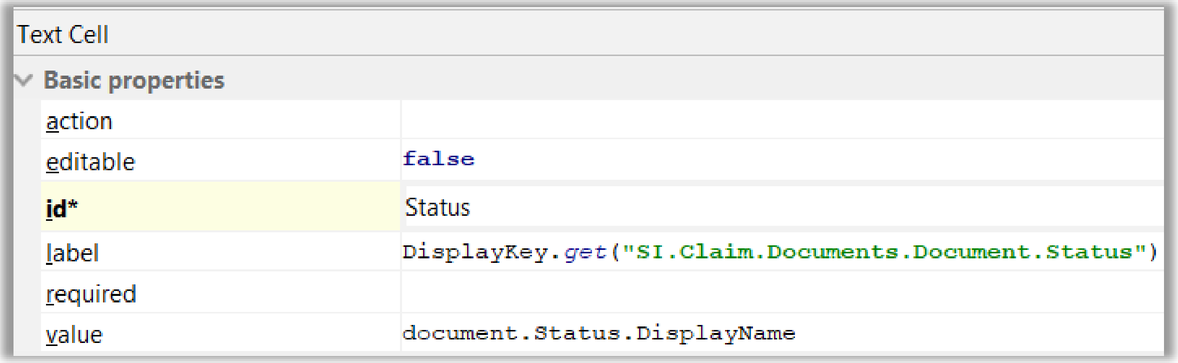


Text Cell

Basic properties

action	
editable	false
id*	Type
label	`DisplayKey.get("SI.Claim.Documents.Document.Type")`
required	
value	`document.Type.DisplayName`

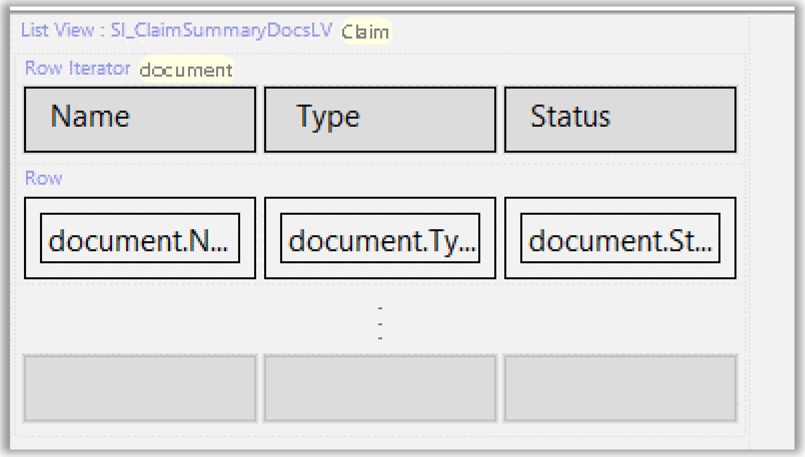
editable	**False**
id	**Status**
label	**DisplayKey.get("SI.Claim.Documents.Document.Status")**
value	**document.Status.DisplayName**



Text Cell

Basic properties

action	
editable	false
id*	Status
label	DisplayKey.get("SI.Claim.Documents.Document.Status")
required	
value	document.Status.DisplayName



List View : SI_ClaimSummaryDocsLV Claim

Row Iterator document

Name	Type	Status
document.N...	document.Ty...	document.St...

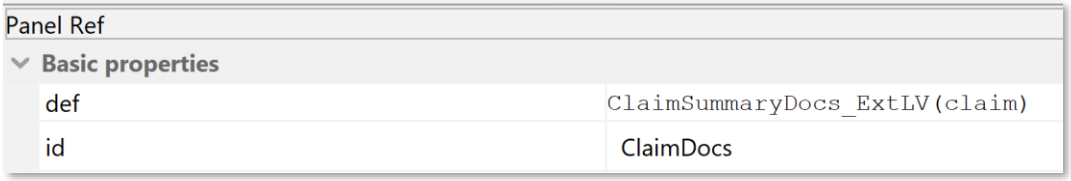
3. **Add the List View you created to the Claim Summary Page.**

   a) Navigate to the **ClaimSummary** Page.

   b) At the bottom of the page, add a **Panel Ref.**

   c) Set its properties as follows:

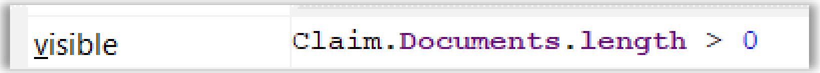
def	**ClaimSummaryDocs_ExtLV(Claim)**
id	**ClaimDocs**
visible	**Claim.Documents.length > 0**



Panel Ref

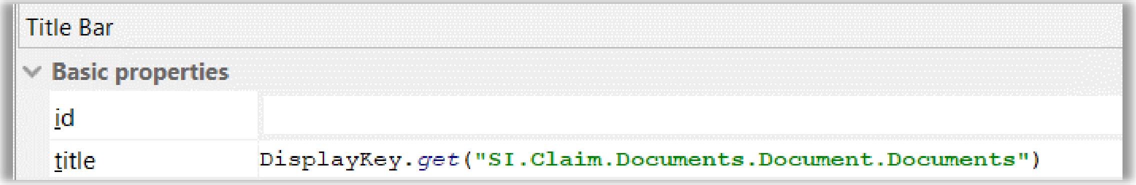
Basic properties

def	ClaimSummaryDocs_ExtLV(claim)
id	ClaimDocs



visible	Claim.Documents.length > 0

   d) Add a Title Bar above the Panel Ref:

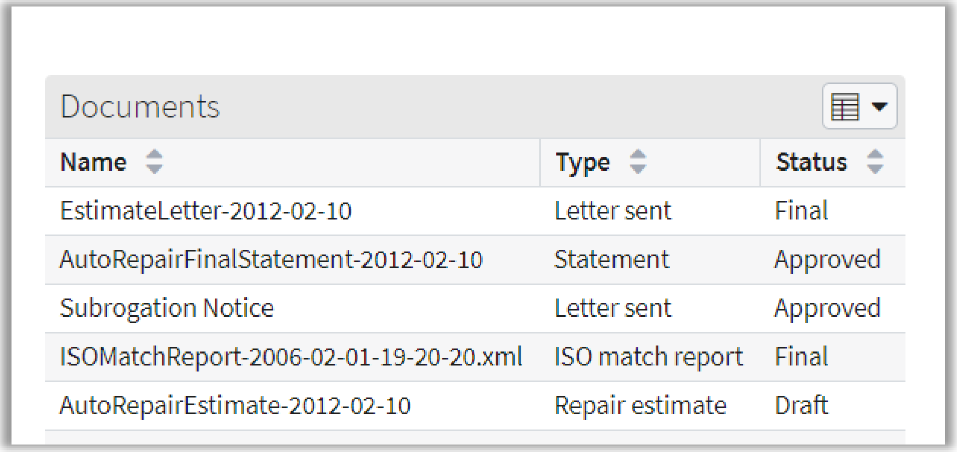


Title Bar

Basic properties

id	
title	`DisplayKey.get("SI.Claim.Documents.Document.Documents")`

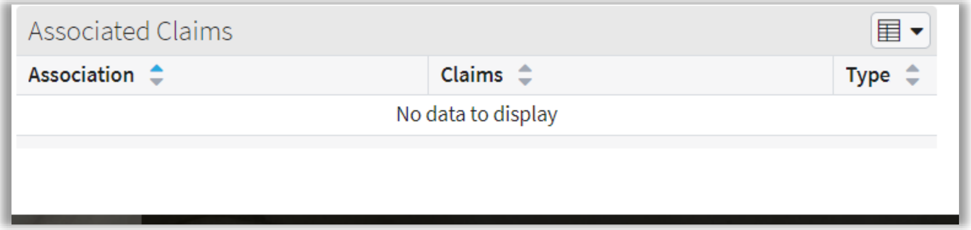
4. **Verify your work**
   a) Switch to a browser open to ClaimCenter.
   b) Type **Alt+Shif+L** to reload the PCF files.
   c) If necessary, log in as **aapplegate/gw** and navigate to the **Ray Newton** claim.
   d) Scroll to the bottom and verify that the Documents list appears as expected.



Documents

Name	Type	Status
EstimateLetter-2012-02-10	Letter sent	Final
AutoRepairFinalStatement-2012-02-10	Statement	Approved
Subrogation Notice	Letter sent	Approved
ISOMatchReport-2006-02-01-19-20-20.xml	ISO match report	Final
AutoRepairEstimate-2012-02-10	Repair estimate	Draft

   e) Navigate to the **Bill Kinman** claim, which has no associated documents, and verify that the Documents list does not appear.

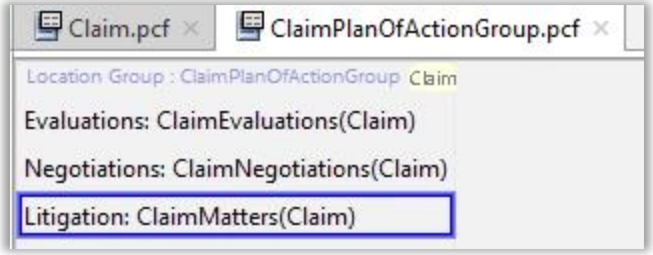


Associated Claims

Association	Claims	Type
	No data to display	

1.3.2 Adding a page to a location group

1. **Determine the PCF elements involved.**
   a) In Guidewire Studio, navigate to the **Claim** Location Group.
   b) Right-click the **Litigation** Location Ref and select **Cut**.
   c) Find the Claim Plan of Action group, note its name, and navigate to it.
   d) Right-click and select **Paste**.
   e) Paste the Litigation location ref at the bottom of the location group.
   f) Delete the location ref's **icon** property.



Location Group : ClaimPlanOfActionGroup Claim

Evaluations: ClaimEvaluations(Claim)

Negotiations: ClaimNegotiations(Claim)

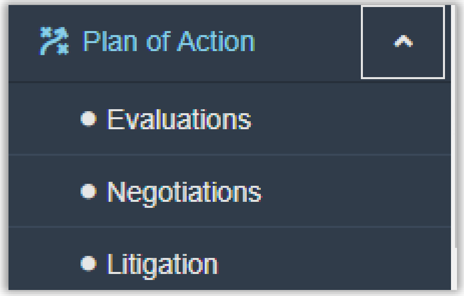
Litigation: ClaimMatters(Claim)

2. **Verify your work.**

   a) In ClaimCenter, reload the PCF files.

   b) Navigate to the Summary Page of any Claim.

   c) Verify that the Litigation menu link appears at the bottom of the Plan of Action group, and ***only*** there.



Lesson 2

Configuring Lines of Business

ClaimCenter's products must match the actual products sold by your company.

The Line of Business (LOB) model provides the flexibility to mirror any insurer's product line faithfully.

2.1 Exploring line of business typelists

Prerequisites

For this lab, you will use ClaimCenter, Guidewire Studio, and a supported web browser.

ClaimCenter Server should be running in debug mode when you start the lab.

The default URL for ClaimCenter is **http://localhost:8080/cc/ClaimCenter.do**.

2.1.1 Lab

Activity

1. **Explore the Line of Business model in the Data Dictionary.**
   a) Open the Data Dictionary.
   b) List the child or children of CoverageType Bobtail Liability (typecode **BABobtailLiabCov**):

   c) List the parent or parents of ExposureType Employer Liability (typecode **EmployerLiability):**

   d) Find a path from LossType Auto (typecode **Auto**) to ExposureType Bodily Injury (typecode **BodilyInjuryDamage**). (There may be more than one path.)

STOP

2.2 Configuring lines of business

Requirement 1: Succeed Insurance wants the Homeowner's Earthquake Coverage (code **HODW_Earthquake**) to map to three exposure types: **Dwelling**, **Other Structure**, and **Content**.

However, in the base configuration, **HODW_Earthquake** has only one child, **General Damage**. You must make the appropriate changes in the LOB model.

Requirement 2: Succeed Insurance offers erosion coverage with Homeowner's policies. Erosion exposures do not exist in the base configuration. You must create the appropriate exposure and mappings.

2.2.1 Lab

Activity

1. **Requirement 1**

    a) In Studio, navigate to the **CoverageType** typelist.
    b) Locate the Homeowner's Earthquake coverage (code **HOPEarthquake**).
    c) Retire the existing **HOPEarthquake** coverage subtype.
    d) Add a coverage subtype that maps to the **Content** Exposure.
    e) Add a coverage subtype that maps to the **Dwelling** Exposure.
    f) Add a coverage subtype that maps to the **Other Structure** Exposure.

2. **Requirement 2**

    a) In Studio, navigate to the **ExposureType** typelist.
    b) Create an ExposureType for **Erosion**. Its Incident should be **FixedPropertyIncident**.
    c) Navigate to configuration ➔ gsrc ➔ libraries ➔ ExposureUI.gsx.
    d) In the second ("else if") list of exposure types, copy the **TC_GENERALDAMAGE** line and past the copy immediately before that line.
    e) Change the new line to read **TC_EROSION**.
    f) Create a mode of the NewClaimExposureDV to handle Erosion exposures.
        Note: This is best done by copying the General Damage mode.
    g) Navigate to the **PolicyType** typelist.
    h) Locate and expand the **Homeowners** coverage (code **HOPHomeowners).**
    i) In the Children node, create a new CoverageType called **Erosion**.
    j) Right-click the new CoverageType (in the Children node) and create a new CoverageSubtype called **HOErosion**.
    k) Map the new CoverageSubtype to the Erosion ExposureType.

3. **Test Procedure**

    a) Start (or restart) the ClaimCenter server in Debug mode.
    b) Log in as aapplegate/gw.

c) Create a Homeowner's claim for **Ken Darion**. Make up the details, filling in only required fields.
d) Assign the claim to **aapplegate**.
e) Navigate to the claim that you just created.
f) Verify that you can add Earthquake exposures for Content, Dwelling, and Other Structure.
g) Verify that you can add an Erosion exposure.

STOP

2.3 Solutions

2.3.1 Exploring line of business typelists

Solution

Exact details on how to complete the lab

1. **Explore the Line of Business model in the Data Dictionary.**
   a) Open the Data Dictionary.
   b) List the child or children of CoverageType Bobtail Liability (typecode **BABobtailLiabCov**):
   3 CoverageSubtypes: BABobtailLiabCov_bi; BABobtailLiabCov_pd; and BABobtailLiabCov_vd

   c) List the parent or parents of ExposureType Employer Liability (typecode **EmployerLiability):**
   2 CoverageSubtypes: FWCFedEmpLiabCov and WCEmpLiabCov

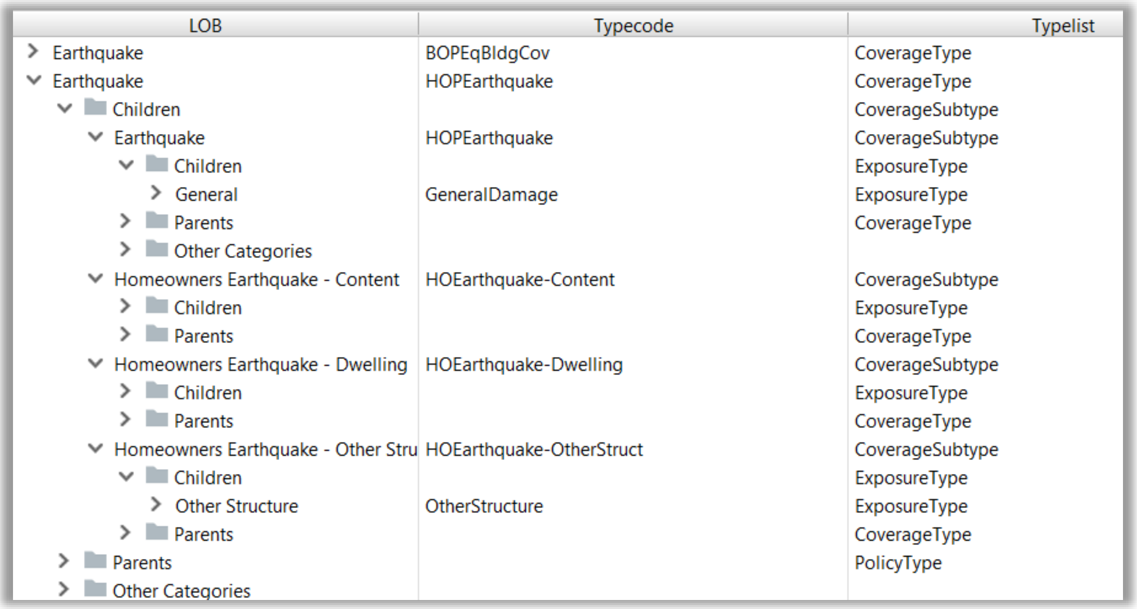
   d) Find a path from LossType Auto (typecode **Auto**) to ExposureType Bodily Injury (typecode **BodilyInjuryDamage**). (There may be more than one path.)
   One possible answer: LossType AUTO ➔ LOBCode PersonalAuto ➔ PolicyType PersonalAuto ➔ CoverageType PADeathDisabilityCov ➔ CoverageSubtypePADeathDisabilityCov ➔ ExposureType BodilyInjuryDamage

2.3.2 Configuring lines of business

Solution

Exact details on how to complete the lab.

1. **Requirement 1**
   a) In Studio, navigate to the **CoverageType** typelist.
   b) Locate the Homeowner's Earthquake coverage (code **HOPEarthquake**).
   c) Add a coverage subtype that maps to the **Content** Exposure.
   d) Add a coverage subtype that maps to the **Dwelling** Exposure.
   e) Add a coverage subtype that maps to the **Other Structure** Exposure.

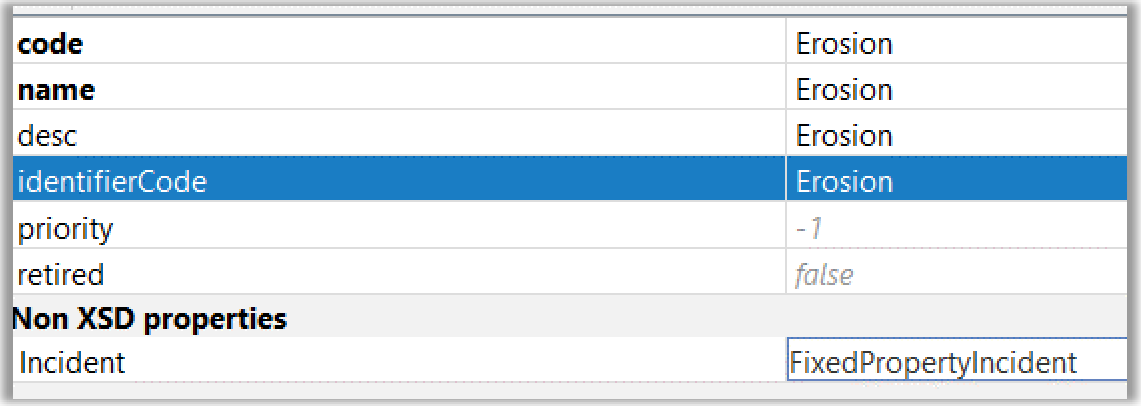


LOB	Typecode	Typelist
> Earthquake	BOPEqBldgCov	CoverageType
∨ Earthquake	HOPEarthquake	CoverageType
∨ Children		CoverageSubtype
∨ Earthquake	HOPEarthquake	CoverageSubtype
∨ Children		ExposureType
> General	GeneralDamage	ExposureType
> Parents		CoverageType
> Other Categories		
∨ Homeowners Earthquake - Content	HOEarthquake-Content	CoverageSubtype
> Children		ExposureType
> Parents		CoverageType
∨ Homeowners Earthquake - Dwelling	HOEarthquake-Dwelling	CoverageSubtype
> Children		ExposureType
> Parents		CoverageType
∨ Homeowners Earthquake - Other Stru	HOEarthquake-OtherStruct	CoverageSubtype
∨ Children		ExposureType
> Other Structure	OtherStructure	ExposureType
> Parents		CoverageType
> Parents		PolicyType
> Other Categories		

2. Requirement 2

a) In Studio, navigate to the **ExposureType** typelist.

b) Create an ExposureType for **Erosion**. Its Incident should be **FixedPropertyIncident**.

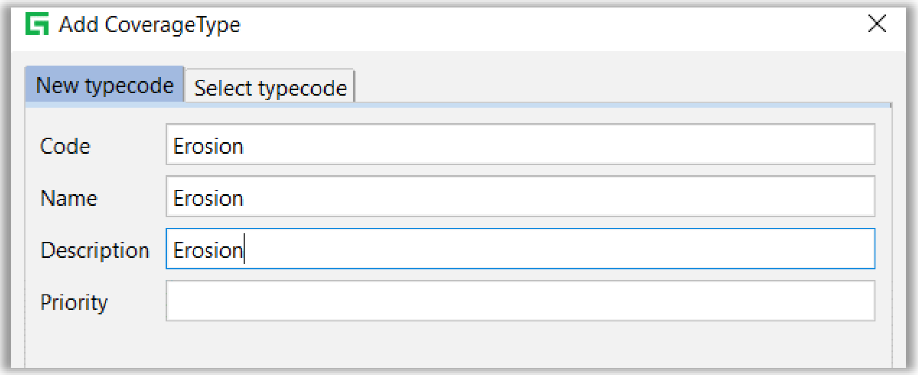


code	Erosion
name	Erosion
desc	Erosion
identifierCode	Erosion
priority	*-1*
retired	*false*
Non XSD properties	
Incident	FixedPropertyIncident

c) Navigate to the **PolicyType** typelist.

d) Locate and expand the **Homeowners** coverage (code **HOPHomeowners).**

e) In the Children node, create a new CoverageType called **Erosion**.

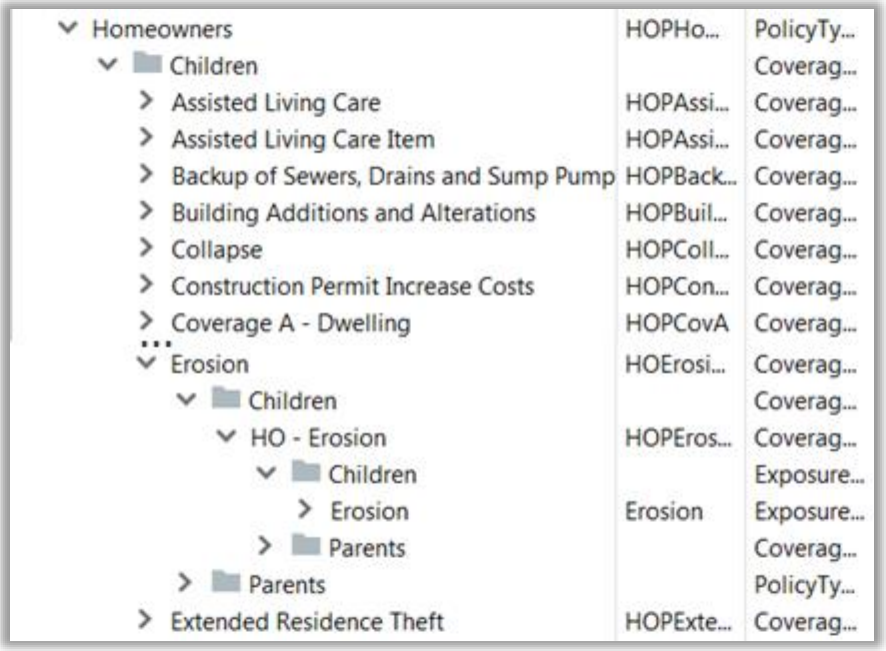


Add CoverageType

Code	Erosion
Name	Erosion
Description	Erosion
Priority	

f) Right-click the new CoverageType (in the Children node) and create a new CoverageSubtype called **HOErosion**.

g) Map the new CoverageSubtype to the Erosion ExposureType.



```
Homeowners                                     HOPHo...    PolicyTy...
  Children                                                 Coverag...
    Assisted Living Care                       HOPAssi...  Coverag...
    Assisted Living Care Item                  HOPAssi...  Coverag...
    Backup of Sewers, Drains and Sump Pump     HOPBack...  Coverag...
    Building Additions and Alterations         HOPBuil...  Coverag...
    Collapse                                   HOPColl...  Coverag...
    Construction Permit Increase Costs         HOPCon...   Coverag...
    Coverage A - Dwelling                      HOPCovA     Coverag...
    ...
    Erosion                                    HOErosi...  Coverag...
      Children                                             Coverag...
        HO - Erosion                           HOPEros...  Coverag...
          Children                                         Exposure...
            Erosion                            Erosion     Exposure...
          Parents                                          Coverag...
      Parents                                              PolicyTy...
    Extended Residence Theft                   HOPExte...  Coverag...
```

3. Test Procedure

a) Start (or restart) the ClaimCenter server in Debug mode.

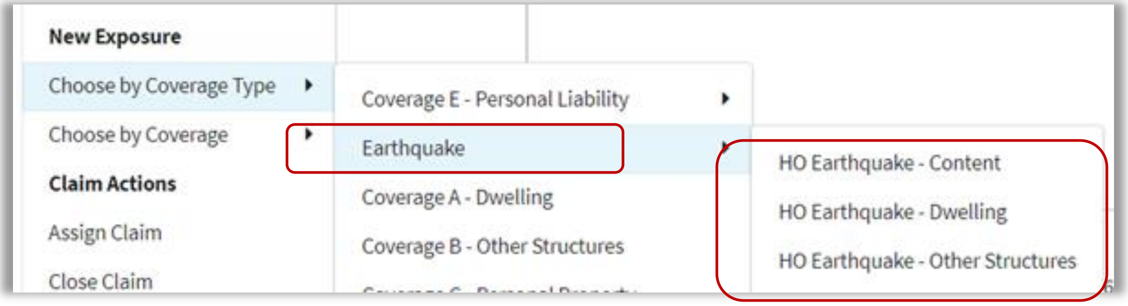
b) Log in as aapplegate/gw.

c) Create a new Homeowner's claim for **Ken Darion**. Make up the details, filling in only required fields.

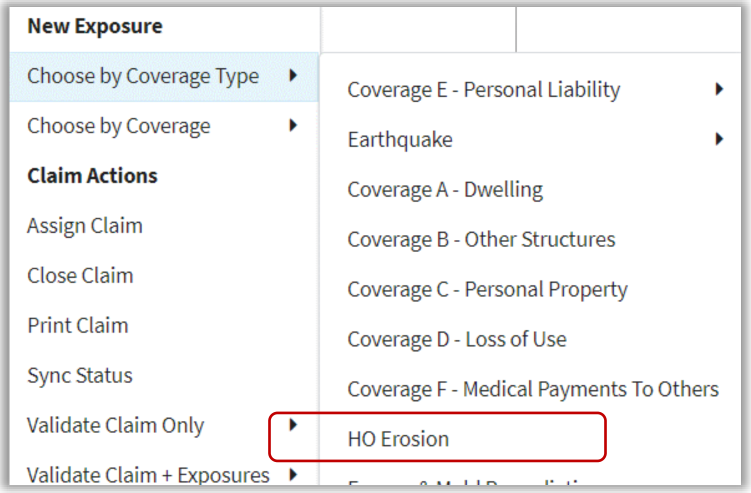
d) Assign the claim to **aapplegate**.

e) Navigate to the claim that you just created.

f) Verify that you can add Earthquake exposures for Content, Dwelling, and Other Structure



g) Verify that you can add an Erosion exposure.



Lesson 3

Configuring Claim Intake

Wizards, including the First Notice of Loss (FNOL) wizard, help make sure that the correct information on a claim is taken in the correct order. You can modify a wizard to better reflect the information your organization gathers and the order in which it is gathered.

3.1 Configuring the FNOL wizard

Prerequisites

For this lab, you will use ClaimCenter, Guidewire Studio, and a supported web browser.

ClaimCenter Server should be running in debug mode when you start the lab.

The default URL for ClaimCenter is **http://localhost:8080/cc/ClaimCenter.do**.

3.1.1 Lab

During claim intake for Personal Auto claims, Succeed Insurance customer service representatives often access the list of involved vehicles several times throughout the intake call.

They want service reps to have an easier time identifying where to access this information in the wizard. Therefore, they are requesting that the list of involved vehicles, currently located on the Basic Info step, be moved to its own Vehicle Info step.

Tip

Before proceeding, read all the steps below to understand the overall process.

Activity

At a high level, you will start with the existing screen and use two copies of the screen to represent 1) the new basic info screen for Personal Auto claims, and 2) the new vehicle info screen for Auto claims. The original screen will still be used for non-Personal Auto claims.

1. **Create a new screen to contain the Vehicle information.**

   Recommended approach:

   - Duplicate (copy) the existing Basic Info screen that contains the Vehicle info
   - Give it a new filename, FNOLWizard_VehicleInfo_ExtScreen.

- In the new VehicleInfo screen, disable the entire left column.

2. **Ensure that the right hand panel of the default screen will not be visible for personal auto claims.**
3. **Create new display keys containing the title and the labels associated with your new Wizard Step.**

   Recommended approach:

   - Locate the display keys used for the Basic Info wizard step.
   - Copy these display keys and modify the copies appropriately

4. **Create a new wizard step immediately after the Basic Information step which is labeled "Vehicle Information" in the sidebar.**

   Recommended approach:

   - Duplicate the Basic Info wizard step and modify it to refer to your new Vehicle Info screen.
   - Clear the **mode** property.
   - Modify the **title** and **label** properties to refer to the display keys you created.

5. **Modify the wizard step so that the Vehicle Information step appears only for Personal Auto claims.**

3.1.2 Test procedure

1. **In ClaimCenter, type Alt+Shift+L to reload PCFs.**
2. **Log on as aapplegate/gw.**
3. **Start a new personal auto claim using any policy (such as 54-123456).**
   a) Verify that the Basic Information step contains no list of vehicles.
   b) Verify that the Vehicle Information step appears as expected. (It may not appear until you get to the Basic Information step.)
   c) Cancel the claim.

4. **Start a new Homeowner's claim using any Homeowner's policy (such as policy 73-300676)).**
   a) Verify that the Vehicle Information step does not appear in the sidebar.
   b) Verify that the property info is still displayed in the right column of the Basic Information step.
   c) Cancel the claim.

5. **Start a new Commercial Auto claim using any Commercial Auto policy (such as policy 64-318764 (Long Road Trucking)).**
   a) Verify that the Vehicle Info step does not appear in the sidebar.
   b) Verify that the Involved Vehicles step appears in the sidebar.
   c) Verify that the Involved Vehicles do display on the right column of the Basic Information step.
   d) Cancel the claim.

3.2 Solutions

Solution

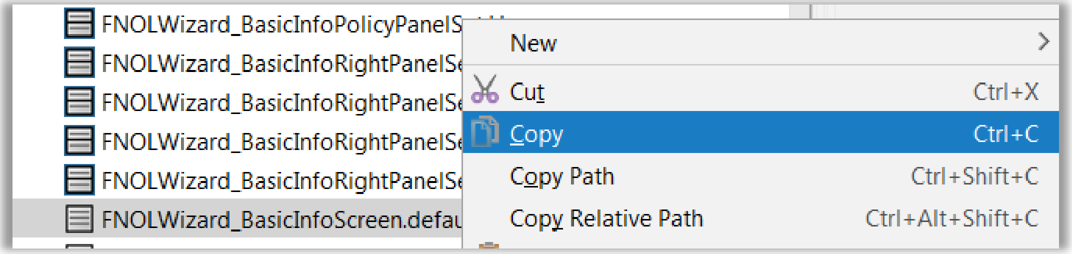
Exact details on how to complete the lab.

3.2.1 Configuring the FNOL wizard

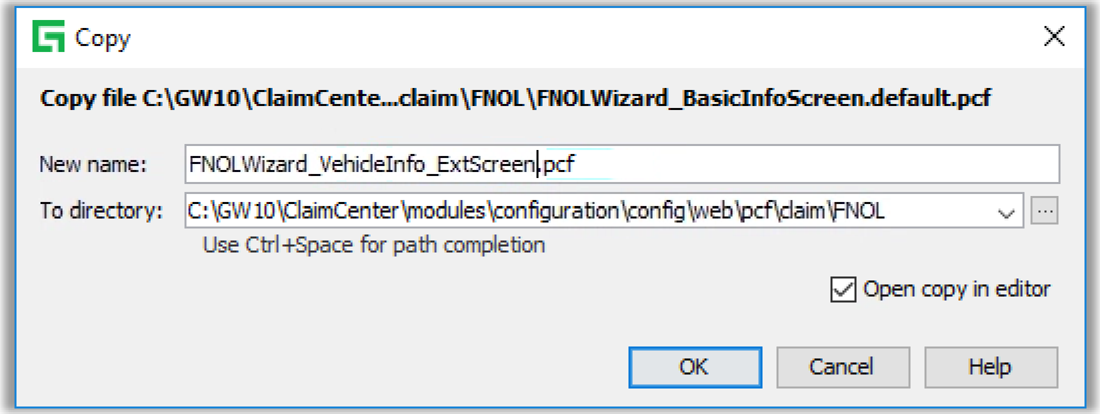
1. **Create a new screen to contain the Vehicle information.**

   Recommended approach:

   - Duplicate (copy) the existing Basic Info screen that contains the Vehicle info


FNOLWizard_BasicInfoPolicyPanelS
FNOLWizard_BasicInfoRightPanelSe
FNOLWizard_BasicInfoRightPanelSe
FNOLWizard_BasicInfoRightPanelSe
FNOLWizard_BasicInfoRightPanelSe
FNOLWizard_BasicInfoScreen.defau
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Relative Path Ctrl+Alt+Shift+C

   - Give it a new filename, FNOLWizard_VehicleInfo_ExtScreen.



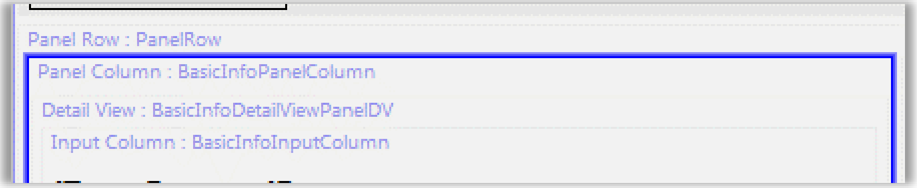
Copy file C:\GW10\ClaimCente...claim\FNOL\FNOLWizard_BasicInfoScreen.default.pcf

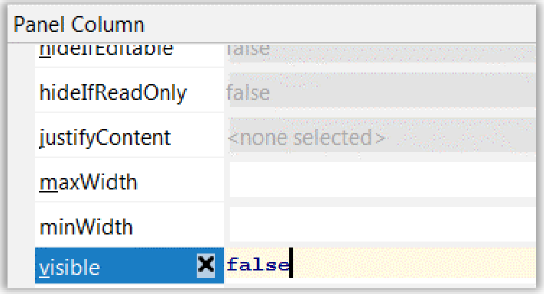
New name: FNOLWizard_VehicleInfo_ExtScreen.pcf

To directory: C:\GW10\ClaimCenter\modules\configuration\config\web\pcf\claim\FNOL

☑ Open copy in editor

   - In the new VehicleInfo screen, disable the entire left column.

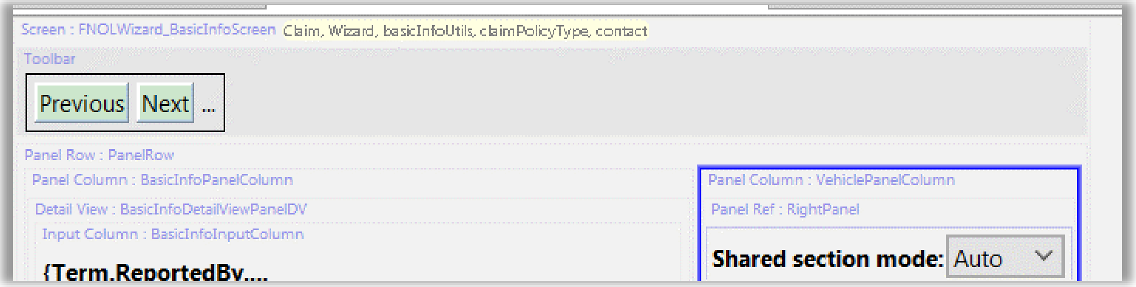

Panel Row : PanelRow
Panel Column : BasicInfoPanelColumn
Detail View : BasicInfoDetailViewPanelDV
Input Column : BasicInfoInputColumn



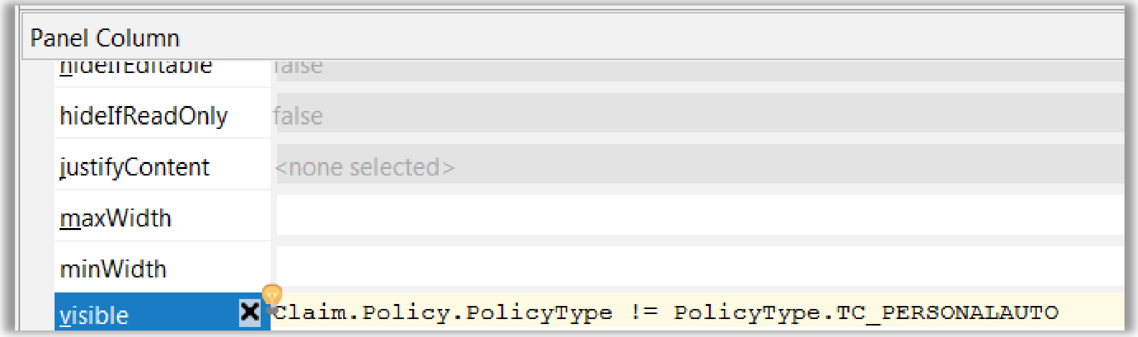
Panel Column

hideIfEditable	false
hideIfReadOnly	false
justifyContent	<none selected>
maxWidth	
minWidth	
visible ☒	false

2. Ensure that the right hand panel of the default screen will not be visible for personal auto claims.


Screen : FNOLWizard_BasicInfoScreen
Claim, Wizard, basicInfoUtils, claimPolicyType, contact
Toolbar
Previous
Next
...
Panel Row : PanelRow
Panel Column : BasicInfoPanelColumn
Detail View : BasicInfoDetailViewPanelDV
Input Column : BasicInfoInputColumn
{Term.ReportedBy....
Panel Column : VehiclePanelColumn
Panel Ref : RightPanel

Shared section mode: Auto



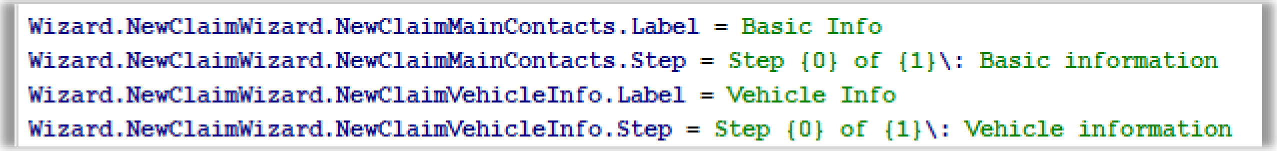
Panel Column

hideIfEditable	false
hideIfReadOnly	false
justifyContent	<none selected>
maxWidth	
minWidth	
visible ☒	Claim.Policy.PolicyType != PolicyType.TC_PERSONALAUTO

3. Create new display keys containing the title and the labels associated with your new Wizard Step.

Recommended approach:

- Locate the display keys used for the Basic Info wizard step.
- Copy these display keys and modify the copies appropriately

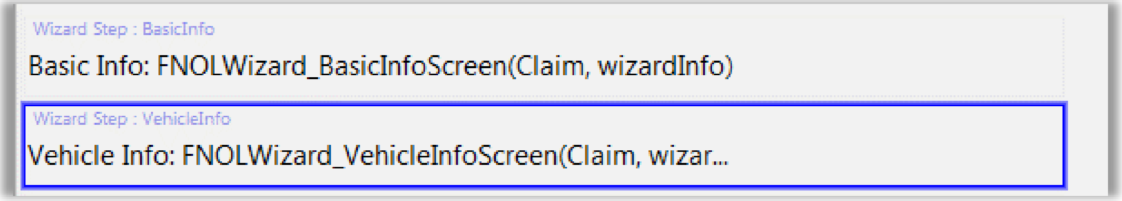


```
Wizard.NewClaimWizard.NewClaimMainContacts.Label = Basic Info
Wizard.NewClaimWizard.NewClaimMainContacts.Step = Step {0} of {1}\: Basic information
Wizard.NewClaimWizard.NewClaimVehicleInfo.Label = Vehicle Info
Wizard.NewClaimWizard.NewClaimVehicleInfo.Step = Step {0} of {1}\: Vehicle information
```

4. **Create a new wizard step immediately after the Basic Information step which is labeled "Vehicle Information" in the sidebar.**

   Recommended approach:

   - Duplicate the Basic Info wizard step and modify it to refer to your new Vehicle Info screen.
   - Clear the **mode** property.
   - Modify the **title** and **label** properties to refer to the display keys you created.

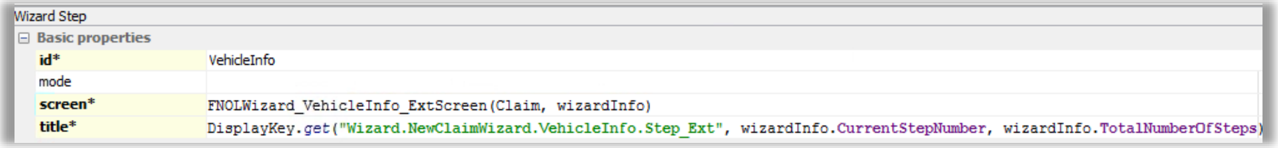


Wizard Step : BasicInfo

Basic Info: FNOLWizard_BasicInfoScreen(Claim, wizardInfo)

Wizard Step : VehicleInfo

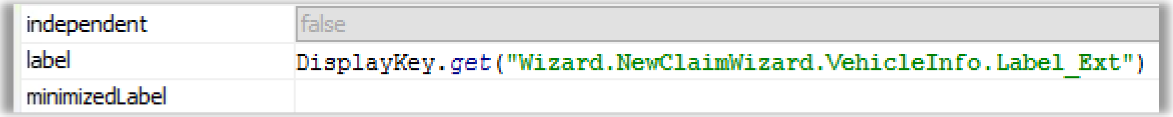
Vehicle Info: FNOLWizard_VehicleInfoScreen(Claim, wizar...



Wizard Step

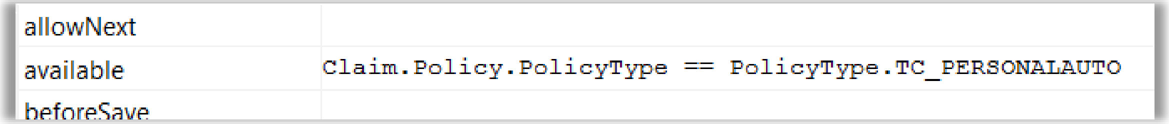
⊟ Basic properties

id*	VehicleInfo
mode	
screen*	FNOLWizard_VehicleInfo_ExtScreen(Claim, wizardInfo)
title*	DisplayKey.get("Wizard.NewClaimWizard.VehicleInfo.Step_Ext", wizardInfo.CurrentStepNumber, wizardInfo.TotalNumberOfSteps)

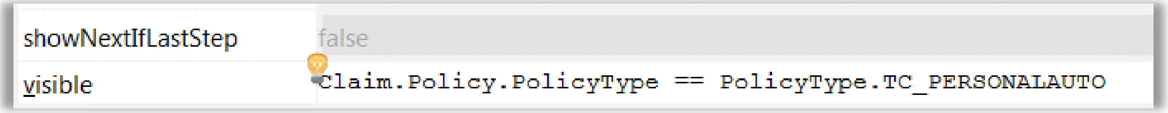


independent	false
label	DisplayKey.get("Wizard.NewClaimWizard.VehicleInfo.Label_Ext")
minimizedLabel	

5. **Modify the wizard step so that the Vehicle Information step appears only for Personal Auto claims.**



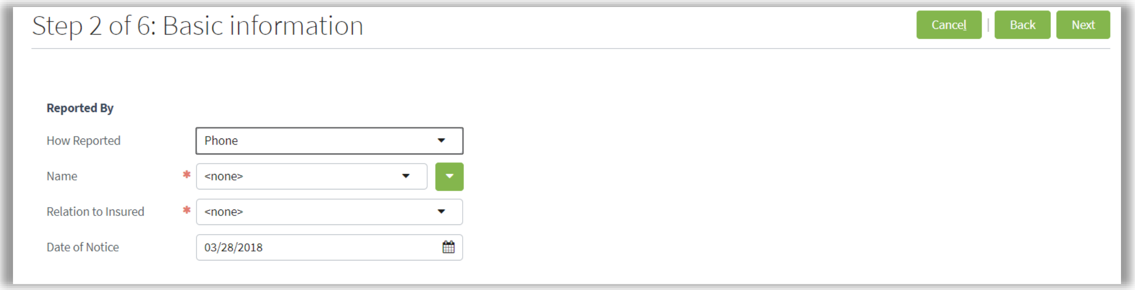
allowNext	
available	Claim.Policy.PolicyType == PolicyType.TC_PERSONALAUTO
beforeSave	



showNextIfLastStep	false
visible	Claim.Policy.PolicyType == PolicyType.TC_PERSONALAUTO

3.2.2 Test procedure

1. **In ClaimCenter, type Alt+Shift+L to reload PCFs.**
2. **Log on as aapplegate/gw.**
3. **Start a new personal auto claim using any policy (such as 54-123456).**
   a) Verify that the Basic Information step contains no list of vehicles.

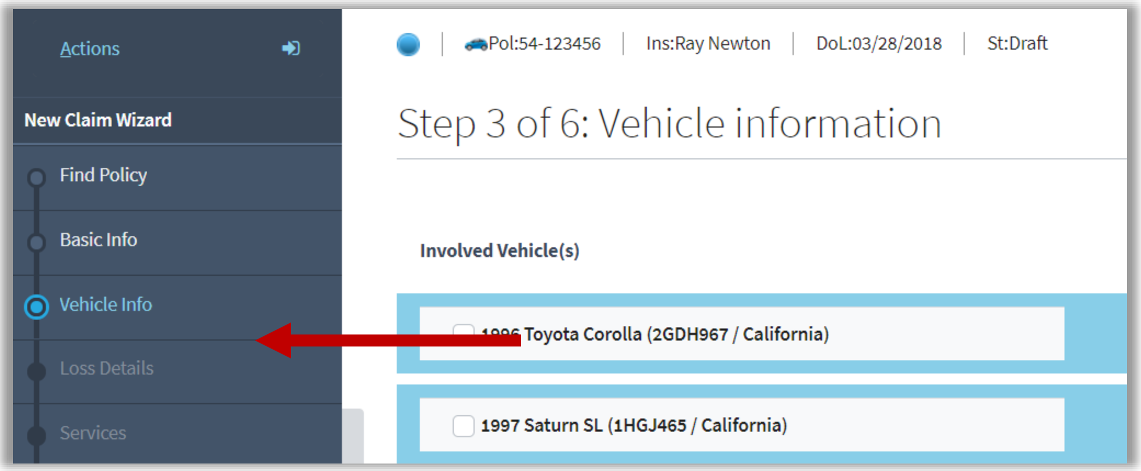


Step 2 of 6: Basic information

Reported By

How Reported	Phone
Name *	<none>
Relation to Insured *	<none>
Date of Notice	03/28/2018

   b) Verify that the Vehicle Information step appears as expected. (It may not appear in the sidebar until you get to the Basic Information step.)



Pol:54-123456 | Ins:Ray Newton | DoL:03/28/2018 | St:Draft

Step 3 of 6: Vehicle information

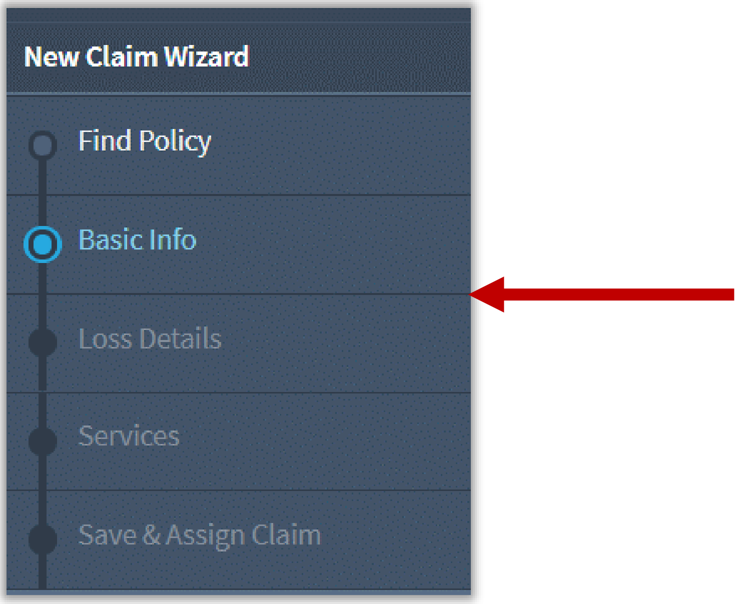
Involved Vehicle(s)

☐ 1996 Toyota Corolla (2GDH967 / California)

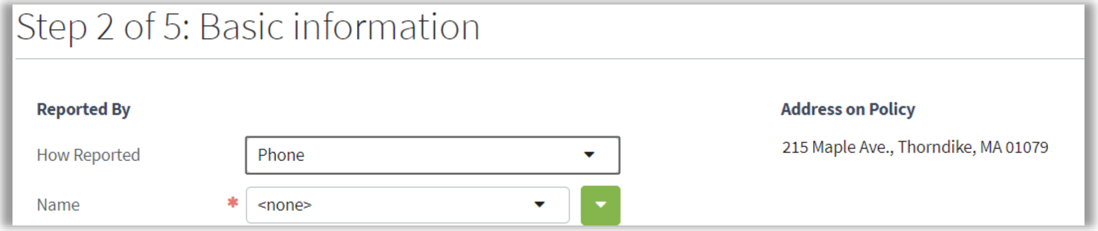
☐ 1997 Saturn SL (1HGJ465 / California)

   c) Cancel the claim.

4. **Start a new Homeowner's claim using any Homeowners policy (such as73-300676).**
   a) Verify that the Vehicle Information step does not appear in the sidebar.



   b) Verify that the property info is still displayed in the right column of the Basic Information step.


Step 2 of 5: Basic information

Reported By

How Reported Phone

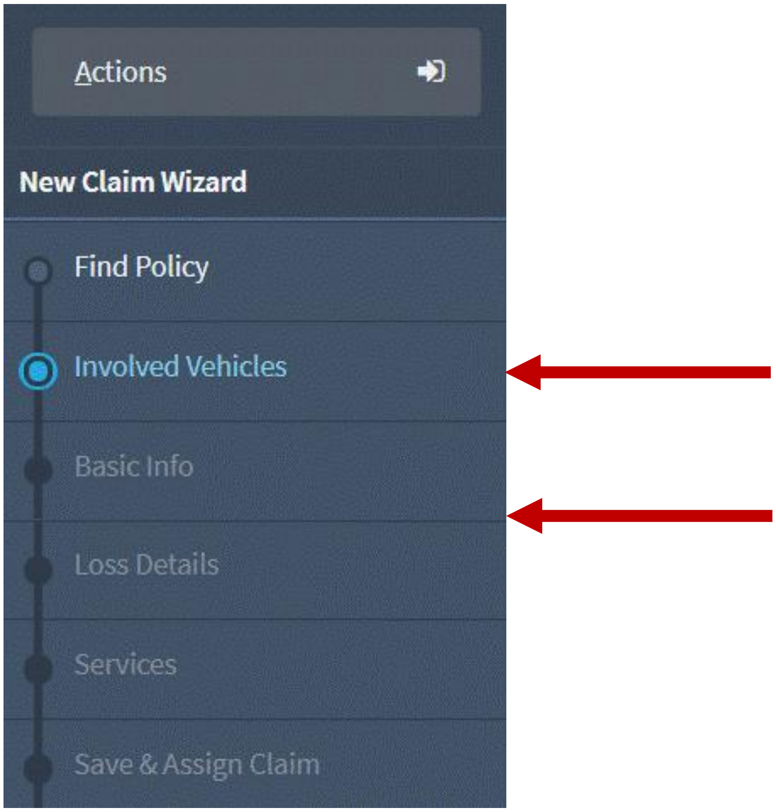
Name * <none>

Address on Policy

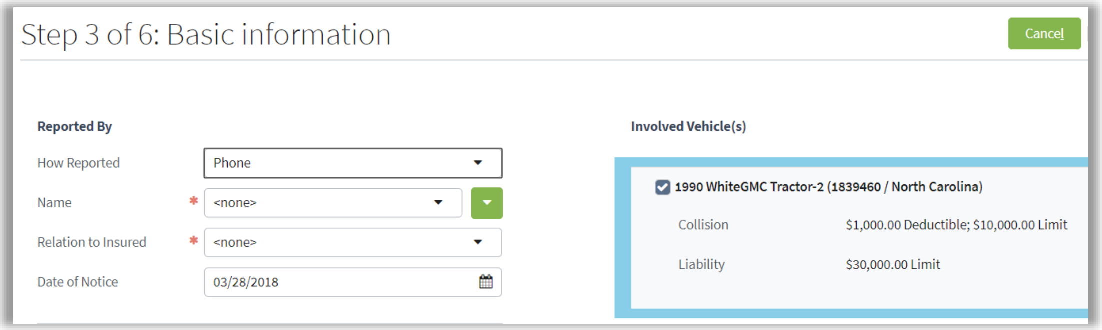
215 Maple Ave., Thorndike, MA 01079

   c) Cancel the claim.

5. **Start a new Commercial Auto claim using any Commercial Auto policy (such as policy 64-318764 (Long Road Trucking)).**
   a) Verify that the Vehicle Info step does not appear in the sidebar.
   b) Verify that the Involved Vehicles step does appear in the sidebar.



   c) Verify that the Involved Vehicles do display on the right column of the Basic Information step.



Step 3 of 6: Basic information

Reported By

How Reported	Phone
Name *	<none>
Relation to Insured *	<none>
Date of Notice	03/28/2018

Involved Vehicle(s)

☑ 1990 WhiteGMC Tractor-2 (1839460 / North Carolina)

Collision	$1,000.00 Deductible; $10,000.00 Limit
Liability	$30,000.00 Limit

   d) Cancel the claim.

Lesson 4

Writing Gosu Rules

As a developer, you must configure ClaimCenter's behavior to match the business logic of a specific insurer.

Gosu rules are a flexible way to configure many different behaviors in the Claim life cycle.

4.1 Create a claim exception rule

Succeed Insurance requires a coroner's report if a fatality has occurred on a Workers Comp claim.

To implement this, an activity to get the coroner's report must be assigned to the user with primary responsibility for the claim.

Prerequisites

For this lab, you will use ClaimCenter, Guidewire Studio, and a supported web browser.

ClaimCenter Server should be running in debug mode when you start the lab.

The default URL for ClaimCenter is **http://localhost:8080/cc/ClaimCenter.do**.

4.1.1 Lab

Cookbook Recipe

Steps to create a new rule

1. **Navigate to the appropriate Rule Set and create a new Rule.**
2. **Set the Rule Condition.**
3. **Set the Rule Action.**

Tip

How can I write the code for these rules?

Some Gosu features that might be useful to you:

- Activity has a method **assign(Group, User)**.

- You can find the Activity Pattern you created by using **ActivityPattern.finder.getActivityPatternByCode(code)**.

Activity

1. **Create a new Activity Pattern.**
   a) Log in to ClaimCenter 10 as **su/gw**.
   b) Navigate to **Administration** ➔ **Business Settings** ➔ **Activity Patterns**.
   c) Create a new Activity Pattern.
   d) Call the Activity Pattern **Require coroner's report when a fatality occurs**.
   e) Set its Category to **Reminder** and its Priority to **Urgent.**
   f) Set its Code to **get_coroner_report**.
   g) Have the activity escalate if it is not completed in **3 business days**.
   h) Set other properties as appropriate and click **Update**.
   i) Log out of ClaimCenter.

2. **Create the new Rule.**
   a) In Guidewire 10 Studio, navigate to the Claim Exception Rules.
   b) Create a Claim Exception Rule called **CER01010 – WC Fatality**.
   c) Place it immediately below CER01000 – Setting SIU Life Cycle State.

3. **Set the Rule Condition.**
   a) What are the conditions required to activate the rule?
   (Hint: There are two conditions.)

   b) Code each of these conditions as a Gosu statement returning a Boolean value.
   c) Place the Gosu statements in the **return** clause, joined by as many **and**s as necessary.

4. **Set the rule actions.**
   a) What is the Rule supposed to do?

   b) Write Gosu code that accomplishes this.

5. **Verify your work.**
   a) Start (or restart) the ClaimCenter server in Debug mode.
   b) Log in to ClaimCenter as **aapplegate/gw**.
   c) Create a Workers' Compensation claim including a fatality. Assign it to **aapplegate**.

d) Log out, then log in as **su/gw**.

e) Type **Alt-Shift-T**.

f) Locate the Claim Exception batch process and click **Run**.

g) Log out, then log in as **aapplegate/gw**.

h) In the Workplan, verify that a “Require coroner’s report” Activity appears.

STOP

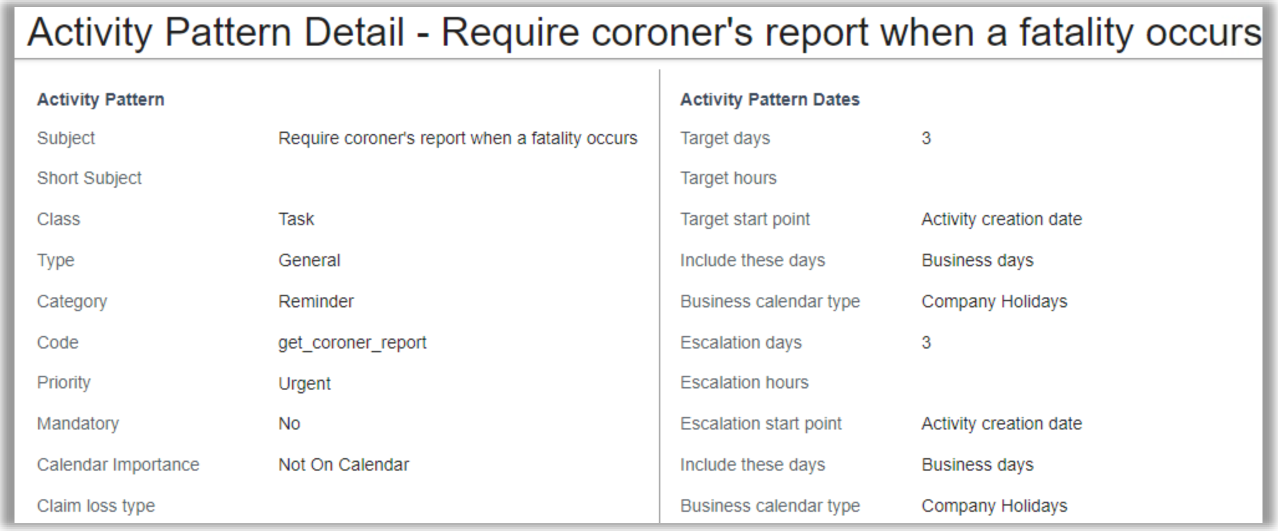
4.2 Solution

4.2.1 Create a claim exception rule

Solution

Exact details on how to complete the lab.

1. **Create a new Activity Pattern.**
   a) Log in to ClaimCenter 10 as **su/gw**.
   b) Navigate to **Administration ➔ Business Settings ➔ Activity Patterns**.
   c) Create a new Activity Pattern.
   d) Call the Activity Pattern **Require coroner's report when a fatality occurs**.
   e) Set its Category to **Reminder** and its Priority to **Urgent.**
   f) Set its Code to **get_coroner_report**.
   g) Have the activity escalate if it is not completed in **3 business days**.
   h) Set other properties as appropriate.



Activity Pattern Detail - Require coroner's report when a fatality occurs

Activity Pattern		Activity Pattern Dates	
Subject	Require coroner's report when a fatality occurs	Target days	3
Short Subject		Target hours	
Class	Task	Target start point	Activity creation date
Type	General	Include these days	Business days
Category	Reminder	Business calendar type	Company Holidays
Code	get_coroner_report	Escalation days	3
Priority	Urgent	Escalation hours	
Mandatory	No	Escalation start point	Activity creation date
Calendar Importance	Not On Calendar	Include these days	Business days
Claim loss type		Business calendar type	Company Holidays

   i) Log out of ClaimCenter.

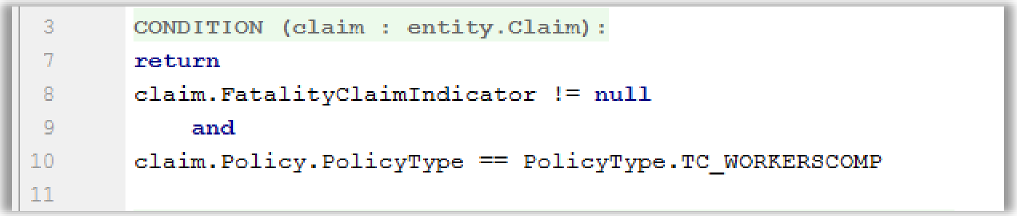
2. **Create the new Rule.**
   a) In Guidewire Studio, navigate to the Claim Exception Rules.
   b) Create a Claim Exception Rule called **CER01010 – WC Fatality**.
   c) Place it immediately below CER01000 – Setting SIU Life Cycle State.

3. **Set the Rule Condition.**
   a) What are the conditions required to activate the rule?
   (Hint: There are two conditions.)

The Claim Fatality indicator field must not be null.
The Policy Type must be Workers' Compensation.

b) Code each of these conditions as a Gosu statement returning a Boolean value.

c) Place the Gosu statements in the **return** clause, joined by as many **and**s as necessary.

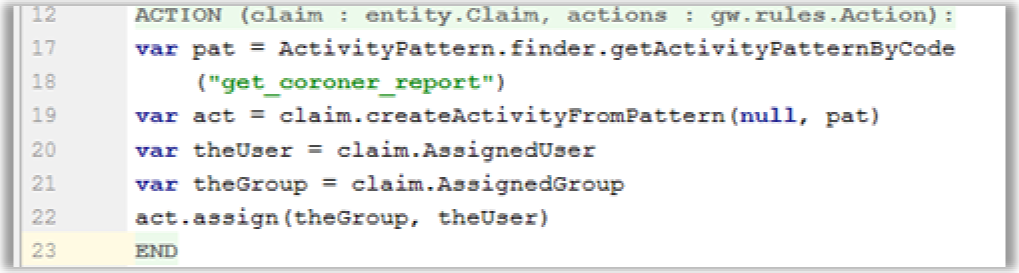


```
CONDITION (claim : entity.Claim):
return
claim.FatalityClaimIndicator != null
    and
claim.Policy.PolicyType == PolicyType.TC_WORKERSCOMP

```

4. Set the rule actions.

a) What is the Rule supposed to do?
Create an activity of type get_coroner_report and assign it to the user assigned to the claim.

b) Write Gosu code that accomplishes this.



```
ACTION (claim : entity.Claim, actions : gw.rules.Action):
var pat = ActivityPattern.finder.getActivityPatternByCode
    ("get_coroner_report")
var act = claim.createActivityFromPattern(null, pat)
var theUser = claim.AssignedUser
var theGroup = claim.AssignedGroup
act.assign(theGroup, theUser)
END
```

5. Verify your work.

a) Start (or restart) the ClaimCenter server.

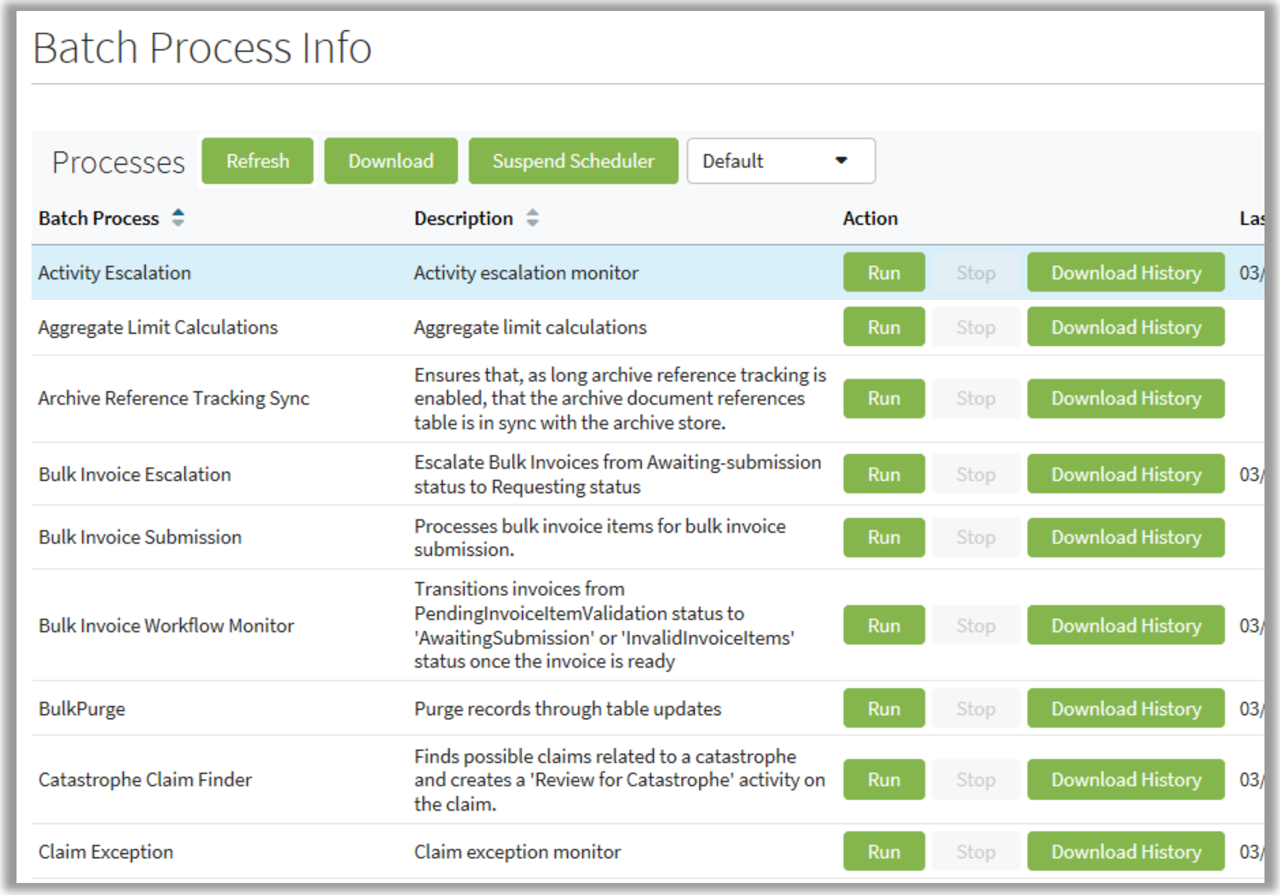
b) Log in to ClaimCenter as **aapplegate/gw**.

c) Create a Workers' Compensation claim including a fatality. Assign it to aapplegate.

d) Log out, then log in as **su/gw**.

e) Type **Alt-Shift-T**.

f) Locate the Claim Exception batch process and click **Run**.



Batch Process Info

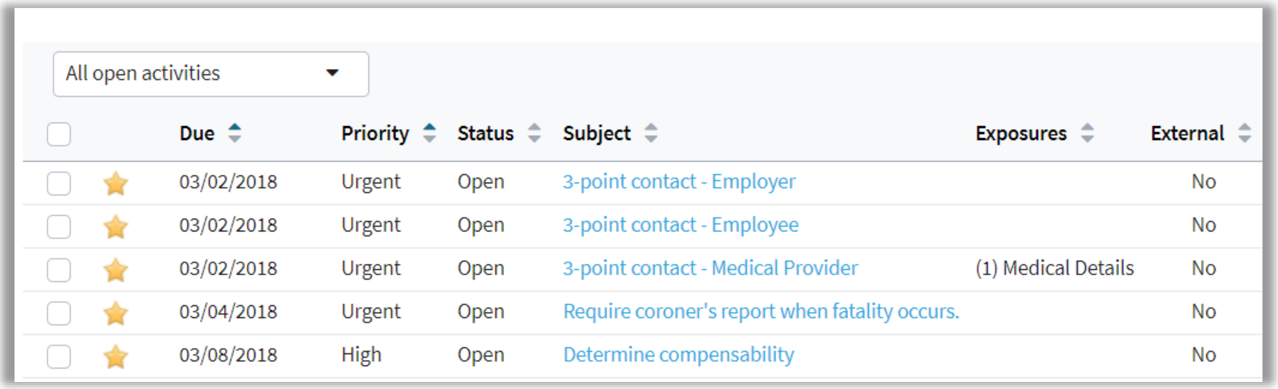
Processes

Batch Process	Description	Action			Las
Activity Escalation	Activity escalation monitor	Run	Stop	Download History	03/
Aggregate Limit Calculations	Aggregate limit calculations	Run	Stop	Download History	
Archive Reference Tracking Sync	Ensures that, as long archive reference tracking is enabled, that the archive document references table is in sync with the archive store.	Run	Stop	Download History	
Bulk Invoice Escalation	Escalate Bulk Invoices from Awaiting-submission status to Requesting status	Run	Stop	Download History	03/
Bulk Invoice Submission	Processes bulk invoice items for bulk invoice submission.	Run	Stop	Download History	
Bulk Invoice Workflow Monitor	Transitions invoices from PendingInvoiceItemValidation status to 'AwaitingSubmission' or 'InvalidInvoiceItems' status once the invoice is ready	Run	Stop	Download History	03/
BulkPurge	Purge records through table updates	Run	Stop	Download History	03/
Catastrophe Claim Finder	Finds possible claims related to a catastrophe and creates a 'Review for Catastrophe' activity on the claim.	Run	Stop	Download History	03/
Claim Exception	Claim exception monitor	Run	Stop	Download History	03/

g) Log out, then log in as **aapplegate/gw**.

h) Navigate to the WC claim you created.

i) In the Workplan, verify that a “Require coroner’s report” Activity appears.



□		Due	Priority	Status	Subject	Exposures	External
□	★	03/02/2018	Urgent	Open	3-point contact - Employer		No
□	★	03/02/2018	Urgent	Open	3-point contact - Employee		No
□	★	03/02/2018	Urgent	Open	3-point contact - Medical Provider	(1) Medical Details	No
□	★	03/04/2018	Urgent	Open	Require coroner's report when fatality occurs.		No
□	★	03/08/2018	High	Open	Determine compensability		No

Lesson 5

Writing Assignment Rules

A big part of an efficient and effective claims process is making sure the right work gets to the right people at the right time.

Assignment rules, a type of Gosu rules, help you ensure that Claims, Exposures, Activities, Matters, Service Requests and Subrogations are routed to the correct worker promptly.

5.1 Assigning auto theft claims by location

Succeed Insurance has two groups which handle auto claims in which the entire car was stolen. Succeed Insurance wants complex theft claims to go to either Complex Thefts - West and Complex Thefts - East, depending on the loss location.

Normally, this would probably be done in conjunction with segmentation. A segmentation rule would segment the claim based on loss cause, and the assignment rule would react to the claim's segment. To simplify the lab, you can write the assignment rule to react directly to the loss cause.

5.1.1 Prerequisites

This lesson, like all the lessons in this class, assumes that you have successfully completed the Configuration Fundamentals course.

This lesson assumes that you have successfully completed the lab for Lesson 2, Writing Gosu Rules.

For this lab, you will use ClaimCenter, Guidewire Studio, and a supported web browser.

ClaimCenter Server should be running in debug mode when you start the lab.

The default URL for ClaimCenter is **http://localhost:8080/cc/ClaimCenter.do**.

5.1.2 Lab

Activity

1. **Import required data**
   a) This requirement implies the existence of the Complex Theft groups and their associated users. Your classroom build includes a file, config_groups_users.csv, in the \admin\bin directory that you will import.
   b) Open a command window at C:\Guidewire\ClaimCenter\admin\bin.
   c) Enter import_tools.bat –import config_groups_users.csv –password gw

d) Wait for the command to finish processing. If there are errors (there should be none), contact your instructor.

2. Implement the requirement

a) Create a Global Claim Assignment rule that implements this requirement.
b) The rule should run before other Global Claim Assignment rules.
c) If it succeeds in making an assignment, no other Global Claim Assignment rules should run.
d) If the loss location is not known, assign to the Complex Theft groups by round robin.

3. Test procedure

a) Start (or restart) the ClaimCenter server in debug mode.
b) Log on to ClaimCenter as aapplegate/gw.
c) Navigate to the Allan Robertson claim and click **Loss Details**.
d) Click **Edit**.
e) Change the Loss Cause dropdown to Theft of entire vehicle.
f) Reassign the claim using assignment rules and verify that the claim goes to Complex Thefts - West.
g) Modify the claim so that the loss location is in an eastern state, such as New York.
h) Reassign the claim using assignment rules and verify that the claim goes to Complex Thefts – East.

STOP

5.2 Assigning FNOL error activities to claim creator

When a claim is first created, an "FNOL snapshot" of the claim is taken. This is a read-only record of the information gathered at the time the claim was created. This is useful when important information about the loss changes during adjudication. (For example, a change of loss cause from "collision while turning left" to "collision with auto" might indicate that the insured is trying to hide being at fault.)

Sometimes, information in an FNOL snapshot varies from later information due to human error. When the insured reports that information in the FNOL snapshot was entered inaccurately, Succeed Insurance wants the adjuster to create a "Verify entry of erroneous FNOL information" activity. This activity should be assigned to the person who created the claim.

In the Succeed Insurance Global Activity Assignment rule set, the last rule is a default rule which assigns activities to the issue owner if no other rules have done so. Consequently, every activity which exits the Global Activity Assignment rule set has been assigned to a group and a user.

5.2.1 Lab

Activity

1. **Create the needed administrative data.**
   a) Log on to ClaimCenter as **su/gw**.
   b) Navigate to Administration ➔ Business Settings ➔ Activity Patterns.
   c) Create a new activity pattern that includes the following criteria:

Subject:	Verify FNOL entry error
Category:	Request
Code:	FNOL_error_Ext
Calendar importance:	High
Target/escalation days	3
Start point	Activity creation date
Include days	Business days
Calendar type	Company Holidays

   d) Log out of ClaimCenter.
   e) Stop the ClaimCenter server.

2. **Create the rule.**
   a) Create an activity assignment rule that assigns any activity based on the "FNOL_error_Ext" pattern to the person who created the claim.
      – Use the claim.CreateUser field.

- Ensure that the rule fires before any other rules which may assign the activity.
- If the claim does not have a CreateUser (may occur with imported claims), then assign the activity to the supervisor of the group which currently owns the claim.
- Use the Global Activity Assignment rule set.

3. Test procedure

a) Start the ClaimCenter server in debug mode.
b) Log in as aapplegate/gw.
c) Using the quick claim wizard, create a personal auto claim.
d) Assign the claim to **Dan Henson**.
e) Log out of ClaimCenter and then log back in as **dhenson/gw**.
f) Create a “Request” activity on the claim using the “Verify FNOL entry error” activity pattern.
g) Verify that the activity is automatically assigned to Andy Applegate.

STOP

5.3 Assigning vandalism claims manually

At Succeed Insurance, the adjuster who can best handle an auto vandalism claim depends on several complex factors, such as the type of damage, the state of the vehicle, police department involvement, and so on. Furthermore, the strategy for vandalism assignment changes somewhat over time.

Succeed Insurance has decided to manually assign all auto claims whose loss cause is "Malicious mischief and vandalism".

5.3.1 Lab

Activity

1. **Create the rule.**

   Create a claim assignment rule that assigns any claim with an assigned group and a loss cause of "Malicious mischief and vandalism" to the supervisor of that group for manual "pending assignment".

2. **Test procedure**

   a) Restart the ClaimCenter server in debug mode.
   b) Log in to ClaimCenter as **aapplegate/gw**.
   c) Navigate to any auto claim and change its loss cause to “Malicious mischief and vandalism.”
   d) Reassign the claim using automated assignment.
   e) Verify that the Adjuster field in the status bar reads “Pending assignment.”

STOP

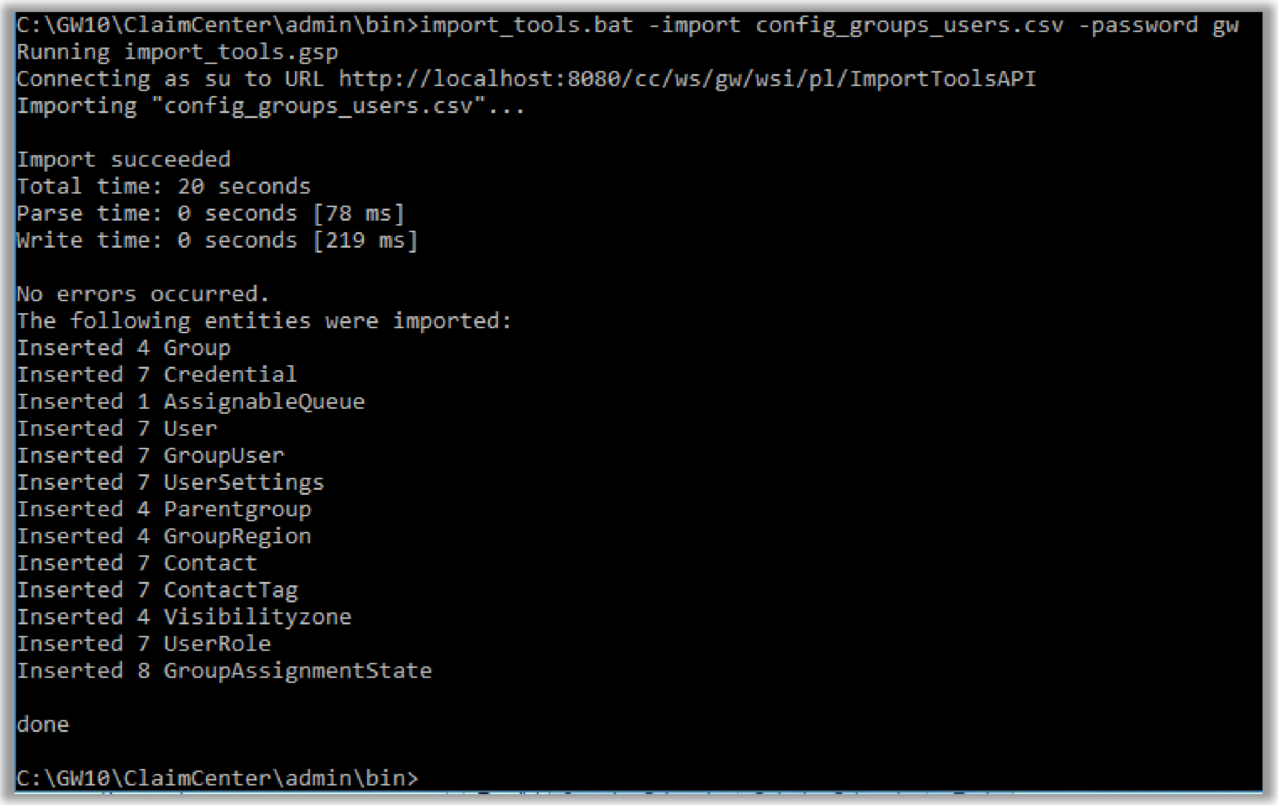
5.4 Solutions

5.4.1 Assigning auto theft claims by location

Solution

Exact details on how to complete the lab.

1. **Import required data**
   a) This requirement implies the existence of the Complex Theft groups and their associated users. Your classroom build includes a file, config_groups_users.csv, in the \admin\bin directory that you will import.
   b) Open a command window at C:\Guidewire\ClaimCenter\admin\bin.
   c) Enter import_tools.bat –import config_groups_users.csv –password gw
   d) Wait for the command to finish processing. If there are errors (there should be none), contact your instructor.

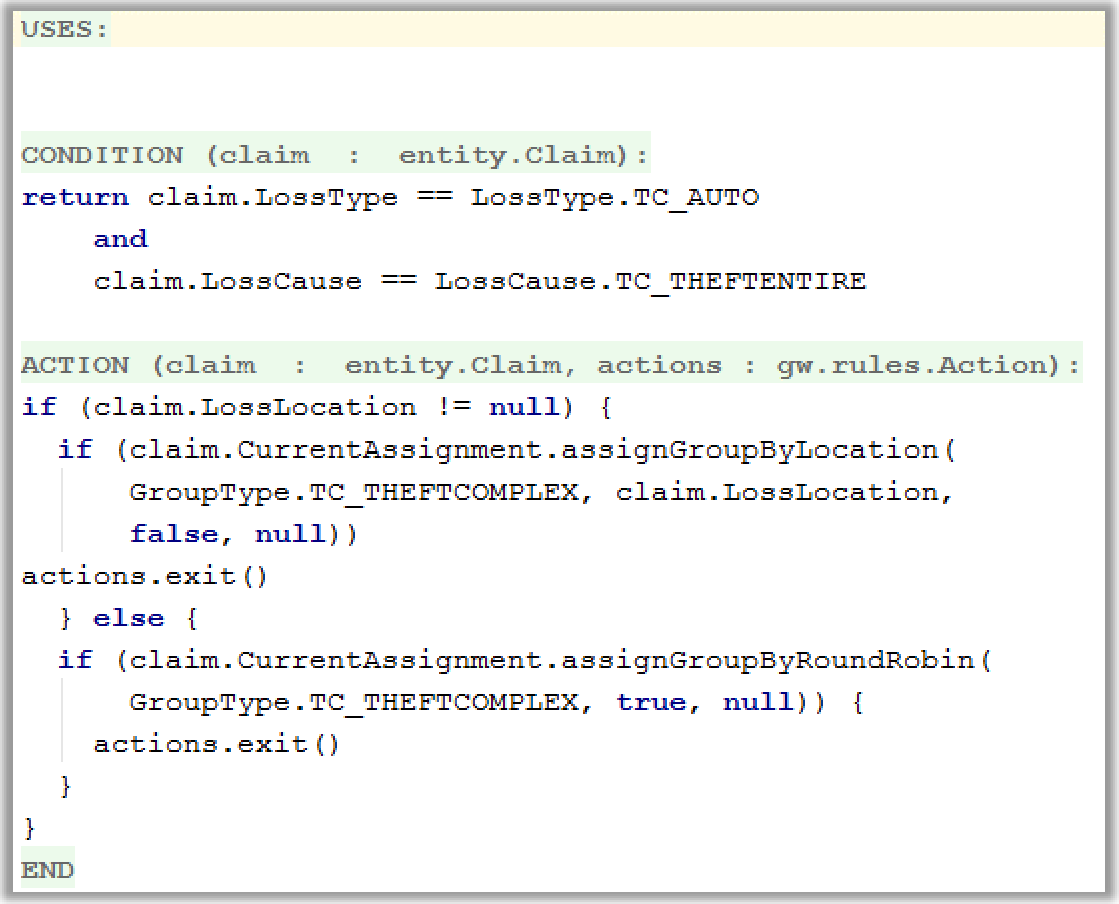


```
C:\GW10\ClaimCenter\admin\bin>import_tools.bat -import config_groups_users.csv -password gw
Running import_tools.gsp
Connecting as su to URL http://localhost:8080/cc/ws/gw/wsi/pl/ImportToolsAPI
Importing "config_groups_users.csv"...

Import succeeded
Total time: 20 seconds
Parse time: 0 seconds [78 ms]
Write time: 0 seconds [219 ms]

No errors occurred.
The following entities were imported:
Inserted 4 Group
Inserted 7 Credential
Inserted 1 AssignableQueue
Inserted 7 User
Inserted 7 GroupUser
Inserted 7 UserSettings
Inserted 4 Parentgroup
Inserted 4 GroupRegion
Inserted 7 Contact
Inserted 7 ContactTag
Inserted 4 Visibilityzone
Inserted 7 UserRole
Inserted 8 GroupAssignmentState

done

C:\GW10\ClaimCenter\admin\bin>
```

2. **Implement the requirement**
   a) Create a Global Claim Assignment rule that implements this requirement.
   b) The rule should run before other Global Claim Assignment rules.

c) If it succeeds in making an assignment, no other Global Claim Assignment rules should run.
d) If the loss location is not known, assign to the Complex Theft groups by round robin.

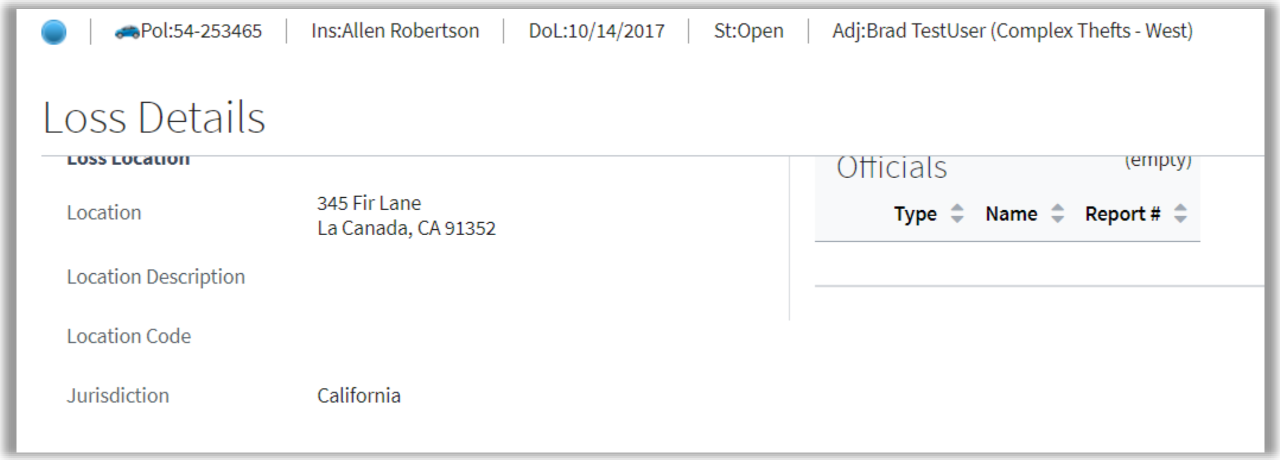
The screenshot shows ***Global Claim Assignment Rule GCA00005 – Auto Theft***



```
USES:


CONDITION (claim  :  entity.Claim):
return claim.LossType == LossType.TC_AUTO
  and
  claim.LossCause == LossCause.TC_THEFTENTIRE

ACTION (claim  :  entity.Claim, actions : gw.rules.Action):
if (claim.LossLocation != null) {
 if (claim.CurrentAssignment.assignGroupByLocation(
   GroupType.TC_THEFTCOMPLEX, claim.LossLocation,
   false, null))
actions.exit()
 } else {
 if (claim.CurrentAssignment.assignGroupByRoundRobin(
   GroupType.TC_THEFTCOMPLEX, true, null)) {
  actions.exit()
 }
}
END
```

3. Test procedure

a) Start (or restart) the ClaimCenter server in debug mode.
b) Log on to ClaimCenter as aapplegate/gw.
c) Navigate to the Allan Robertson claim and click **Loss Details**.
d) Click **Edit**.
e) Change the Loss Cause dropdown to Theft of entire vehicle.
f) Reassign the claim using assignment rules and verify that the claim goes to Complex Thefts - West.



Pol:54-253465 | Ins:Allen Robertson | DoL:10/14/2017 | St:Open | Adj:Brad TestUser (Complex Thefts - West)

Loss Details

Loss Location

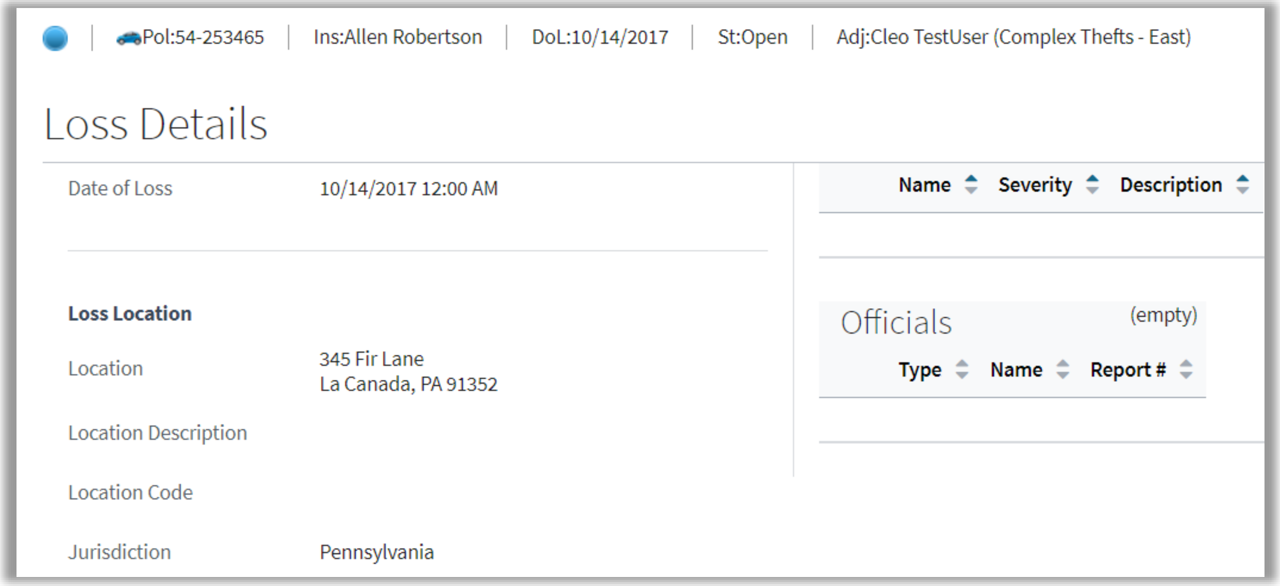
Location	345 Fir Lane La Canada, CA 91352
Location Description	
Location Code	
Jurisdiction	California

Officials (empty)

Type	Name	Report #

g) Modify the claim so that the loss location is in an eastern state, such as New York.

h) Reassign the claim using assignment rules and verify that the claim goes to Complex Thefts – East.



Pol:54-253465 | Ins:Allen Robertson | DoL:10/14/2017 | St:Open | Adj:Cleo TestUser (Complex Thefts - East)

Loss Details

Date of Loss	10/14/2017 12:00 AM

Name	Severity	Description

Loss Location

Location	345 Fir Lane La Canada, PA 91352
Location Description	
Location Code	
Jurisdiction	Pennsylvania

Officials (empty)

Type	Name	Report #

5.4.2 Assigning FNOL error activities to claim creator

Solution

Exact details on how to complete the lab.

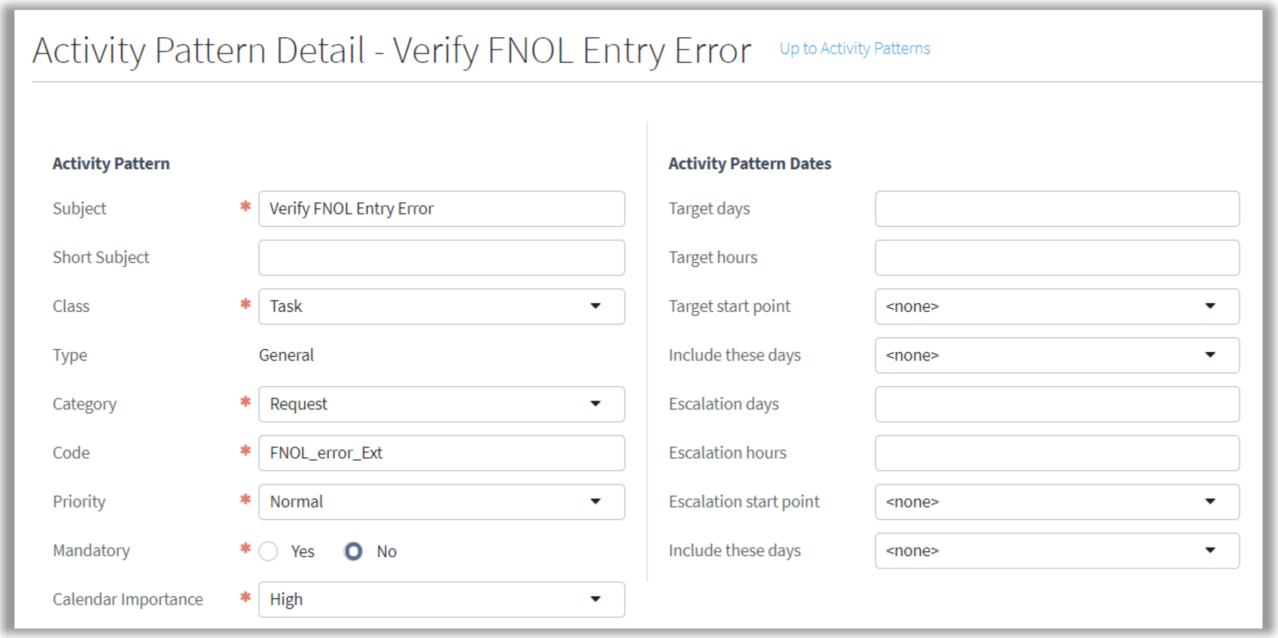
1. **Create the needed administrative data.**

   a) Log on to ClaimCenter as **su/gw**.

   b) Navigate to Administration ➔ Business Settings ➔ Activity Patterns.

   c) Create a new activity pattern that includes the following criteria:

Subject:	Verify FNOL entry error
Category:	Request
Code:	FNOL_error_Ext
Calendar importance:	High



Activity Pattern Detail - Verify FNOL Entry Error

Activity Pattern

Subject	*	Verify FNOL Entry Error
Short Subject		
Class	*	Task
Type		General
Category	*	Request
Code	*	FNOL_error_Ext
Priority	*	Normal
Mandatory	*	○ Yes ● No
Calendar Importance	*	High

Activity Pattern Dates

Target days	
Target hours	
Target start point	<none>
Include these days	<none>
Escalation days	
Escalation hours	
Escalation start point	<none>
Include these days	<none>

d) Log out of ClaimCenter.

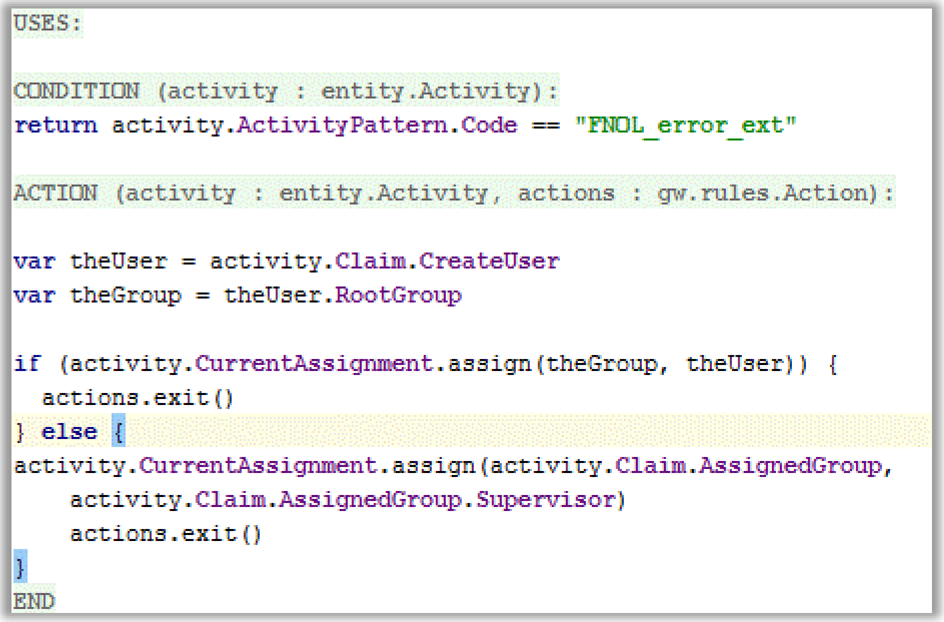
e) Stop the ClaimCenter server.

2. Create the rule.

Create an activity assignment rule that assigns any activity based on the "FNOL_error_Ext" pattern to the person who created the claim.

- Use the claim.CreateUser field.
- Ensure that the rule fires before any other rules which may assign the activity.
- If the claim does not have a CreateUser (may occur with imported claims), then assign the activity to the supervisor of the group which currently owns the claim.
- Use the Global Activity Assignment rule set.

The screenshot shows ***Global Activity Assignment Rule GAA00500 – FNOL entry error****.*



```
USES:

CONDITION (activity : entity.Activity):
return activity.ActivityPattern.Code == "FNOL_error_ext"

ACTION (activity : entity.Activity, actions : gw.rules.Action):

var theUser = activity.Claim.CreateUser
var theGroup = theUser.RootGroup

if (activity.CurrentAssignment.assign(theGroup, theUser)) {
  actions.exit()
} else {
activity.CurrentAssignment.assign(activity.Claim.AssignedGroup,
    activity.Claim.AssignedGroup.Supervisor)
    actions.exit()
}
END
```

3. Test procedure

a) Start the ClaimCenter server in debug mode.

b) Log in as aapplegate/gw.

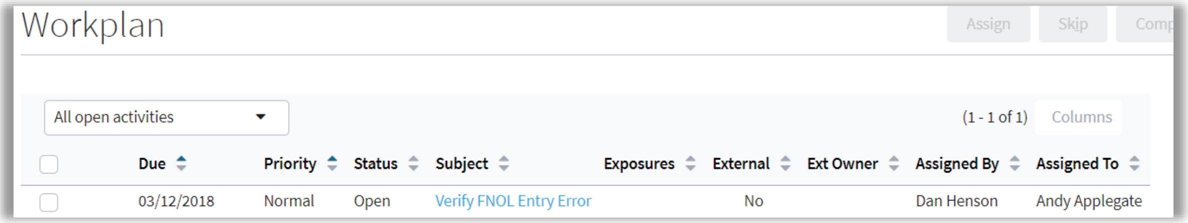
c) Using the quick claim wizard, create a personal auto claim.

d) Assign the claim to **Dan Henson**.

e) Log out of ClaimCenter and then log back in as **dhenson/gw**.

f) Create a “Request” activity on the claim using the “Verify FNOL entry error” activity pattern.

g) Verify that the activity is automatically assigned to Andy Applegate.



Workplan

All open activities ▾

(1 - 1 of 1) Columns

□	Due	Priority	Status	Subject	Exposures	External	Ext Owner	Assigned By	Assigned To
□	03/12/2018	Normal	Open	Verify FNOL Entry Error		No		Dan Henson	Andy Applegate

5.4.3 Assigning vandalism claims manually

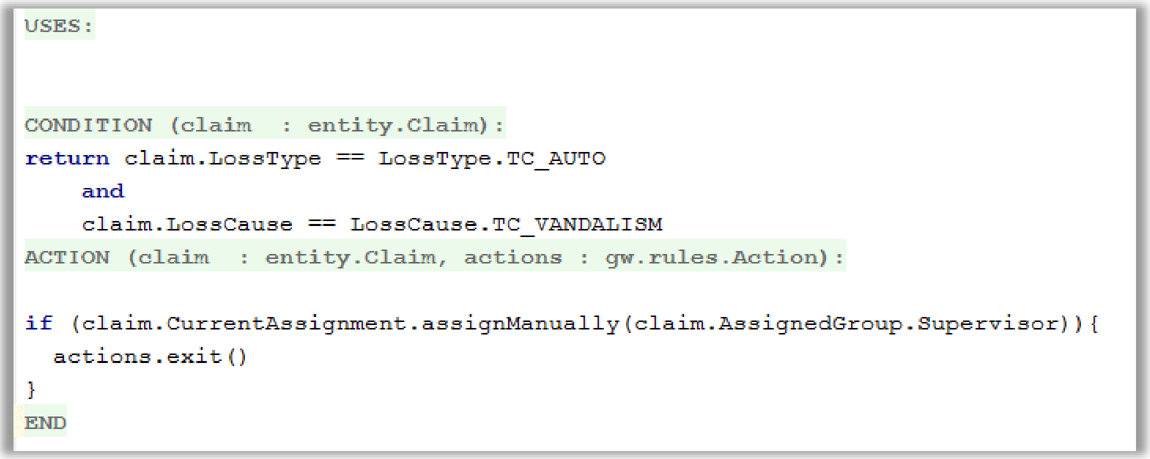
Solution

Exact details on how to complete the lab.

1. Create the rule.

Create a claim assignment rule that assigns any claim with an assigned group and a loss cause of "Malicious mischief and vandalism" to the supervisor of that group for manual "pending assignment".

*The screenshot shows **Default Group Claim Assignment Rule DCG00100 – Auto vandalism.***



```
USES:


CONDITION (claim  : entity.Claim):
return claim.LossType == LossType.TC_AUTO
    and
    claim.LossCause == LossCause.TC_VANDALISM
ACTION (claim  : entity.Claim, actions : gw.rules.Action):

if (claim.CurrentAssignment.assignManually(claim.AssignedGroup.Supervisor)){
  actions.exit()
}
END
```

2. Test procedure

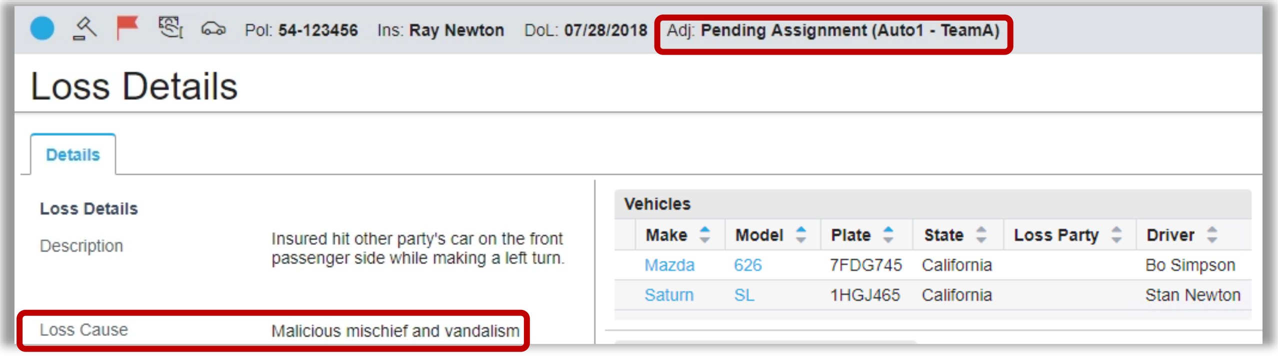
a) Restart the ClaimCenter server in debug mode.

b) Log in to ClaimCenter as **aapplegate/gw**.

c) Navigate to any auto claim and change its loss cause to “Malicious mischief and vandalism.”

d) Reassign the claim using automated assignment.

e) Verify that the Adjuster field in the status bar reads “Pending assignment.”



Pol: **54-123456** Ins: **Ray Newton** DoL: **07/28/2018** Adj: **Pending Assignment (Auto1 - TeamA)**

Loss Details

Details

Loss Details

Description: Insured hit other party's car on the front passenger side while making a left turn.

Loss Cause: Malicious mischief and vandalism

Vehicles

Make	Model	Plate	State	Loss Party	Driver
Mazda	626	7FDG745	California		Bo Simpson
Saturn	SL	1HGJ465	California		Stan Newton

Lesson 6

Writing Claim and Exposure Validation Rules

Claim and Exposure validation rules help ensure that work goes forward, not backward.

6.1 Validation for claim fields

Succeed Insurance has two requirements related to the New Loss level of validation, related to Auto claims.

1. If the loss cause is either "Theft Audio or other parts" or "Theft of entire vehicle", there must be at least one police report listed. (Metro Reports are listed in the Metropolitan Reports list on the Loss Details page.)
2. If the Fault Rating is "no fault" and the loss location's state is California (CA) or Arizona (AZ), warn the user that the given state is not a "no fault" state, but allow the claim be saved. (This can be expanded to other “no fault” states after the rule has been tested.)

In addition, in order for a claim to be valid for referral to the Insurance Services Office (ISO), the loss location must have a postal code.

Prerequisites

This lesson also assumes that you have successfully completed Lesson 2, Writing Gosu rules.

When you start this lab, Studio should be running, and ClaimCenter should be running in debug mode.

The default URL for ClaimCenter is **http://localhost:8080/cc/ClaimCenter.do**.

6.1.1 Lab

Activity

1. **Create display keys.**
   a) Create a display key that rejects a theft claim at New Loss when there is no police report.
   b) Create a display key to report an error when there is no police report on theft of vehicle or audio parts.
   c) Create a display key that rejects a claim if the postal code is not present.
   d) Create a display key to report an error if the postal code is not present.
   e) Create a display key to warn that a given state is not a "no fault" state.
   f) Remember to use the "_Ext" suffix for all display keys you create.

2. **Implement the requirement.**
   a) Create Claim Validation rules that implement this requirement.
   b) The ISO rule should run as a child of the State ISO rule (CLV09800)..
   c) The other two rules can run anywhere in the Claim Validation rules at top level.
   d) For Metro Reports, use the Claim property "MetroReports," which is a list of Metro Reports. (Hint: Use the "length" property of the list.)

6.1.2 Test procedure

1. **Start ClaimCenter in Debug mode.**
2. **Log in as aapplegate/gw.**
3. **Create a new personal auto claim.**
   On the Loss Details step, make sure that:
   - The Postal Code/ZIP Code is left blank.
   - The Loss Cause is set to one of the Theft options.
   - The Loss Location State is set to California or Arizona.
   - There are no Police Reports associated with the claim.
   - The Fault Rating is set to No Fault.

4. **Attempt to save the claim.**
   a) Verify that it cannot be saved due to the missing police report.
   b) Verify that the warning concerning No Fault appears.

5. **Add a police report and attempt to save the claim.**
   Verify that the claim validation level is New Loss Completion.

6. **Using Actions ➔ Validate Claim Only, validate the claim at the New Loss Completion level.**
   Verify that the warning concerning No Fault appears.

7. **Using Actions ➔ Validate Claim Only, validate the claim at the Valid for ISO level.**
   Verify that an error is reported for the missing postal/ZIP code.

8. **Add a postal code to the loss location and save the claim.**

Verify that the claim is saved at the Valid for ISO level or greater.

STOP

6.2 Challenge Requirement: Validation for exposure lists

Complete this task only if time permits, and the instructor tells you to do so.

If an exposure has a listed insurer contact with no phone number, Succeed Insurance does not want the claim to be sent to external systems (validation level “send to external”).

1. **Create a display key to report an error if an exposure has a listed insurer contact, but no phone number for that contact.**
2. **Create the validation rule to meet this requirement.**
3. **Test procedure:**
   Note: To simplify testing, disable the Exposure ISO Validation Rules (EXV05000)
   a) Attempt to create a new collision exposure where there is at least one Other Carrier Involvement with a contact and no phone number.
   b) Verify that the exposure can be saved, but that it cannot reach the Send to external validation level.
   c) Add phone numbers to the carriers with contacts.
   d) Verify that the exposure is now at or beyond the Send to external validation level.
   e) Attempt to remove the phone number from a carrier with a contact, and verify that the change cannot be saved and that the field is flagged.

STOP

6.3 Solutions

6.3.1 Validation for claim fields

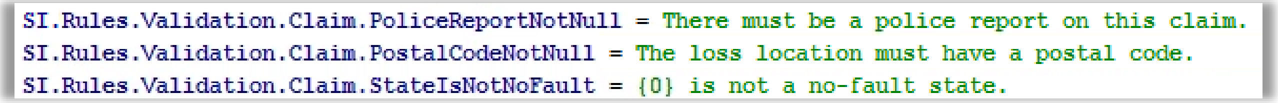
Solution

Exact details on how to complete the lab.

1. **Create display keys.**
   a) Create a display key that rejects a theft claim at New Loss when there is no police report.
   b) Create a display key to report an error when there is no police report on theft of vehicle or audio parts.
   c) Create a display key that rejects a claim if the postal code is not present.

d) Create a display key to warn that a given state is not a “no fault” state.

e) Remember to use an appropriate naming convention for all display keys you create.



```
SI.Rules.Validation.Claim.PoliceReportNotNull = There must be a police report on this claim.
SI.Rules.Validation.Claim.PostalCodeNotNull = The loss location must have a postal code.
SI.Rules.Validation.Claim.StateIsNotNoFault = {0} is not a no-fault state.
```

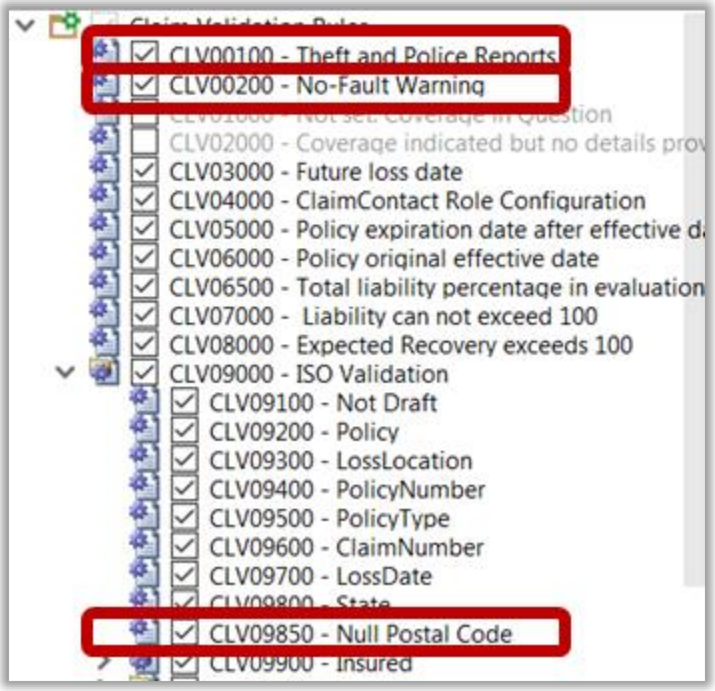
2. Implement the requirement.

a) Create Claim Validation rules that implement this requirement.

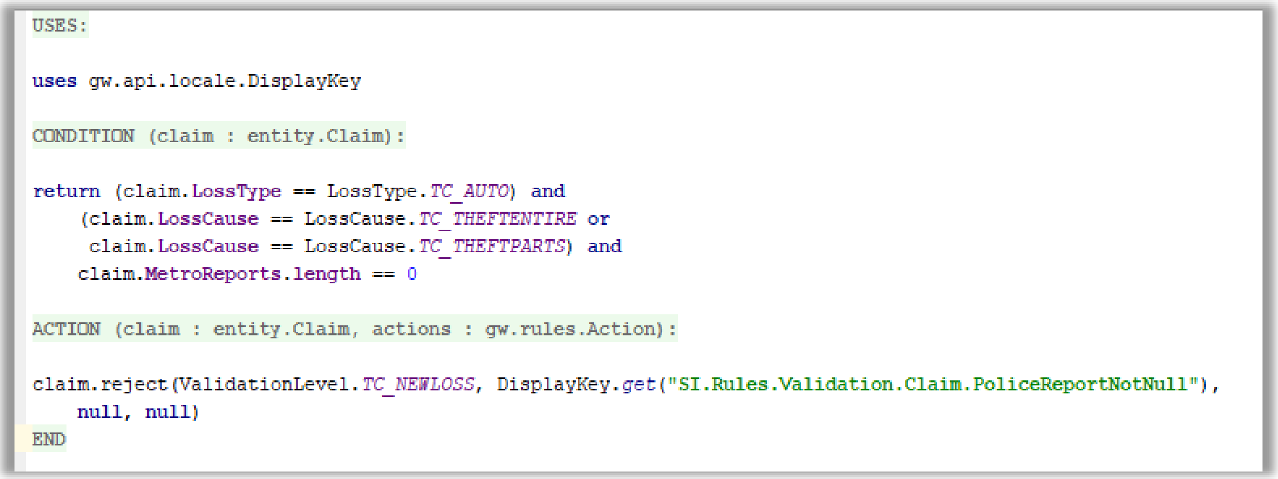
b) The ISO rule should run as a child of the State ISO rule (CLV09800)..

c) The other two rules can run anywhere in the Claim Validation rules at top level.

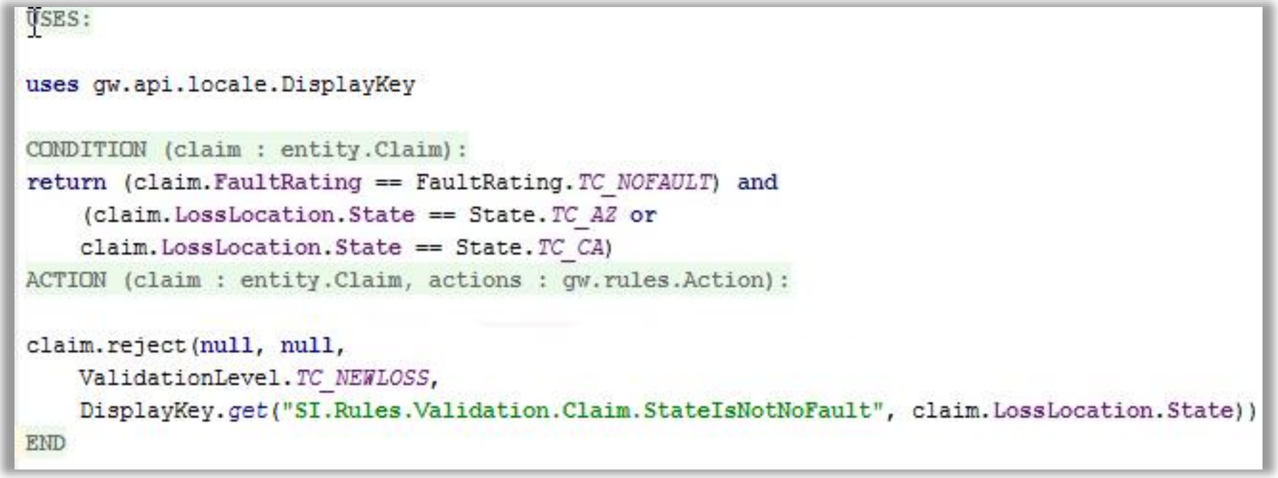
d) For Metro Reports, use the Claim property “MetroReports,” which is a list of Metro Reports. (Hint: Use the “length” property of the list.)



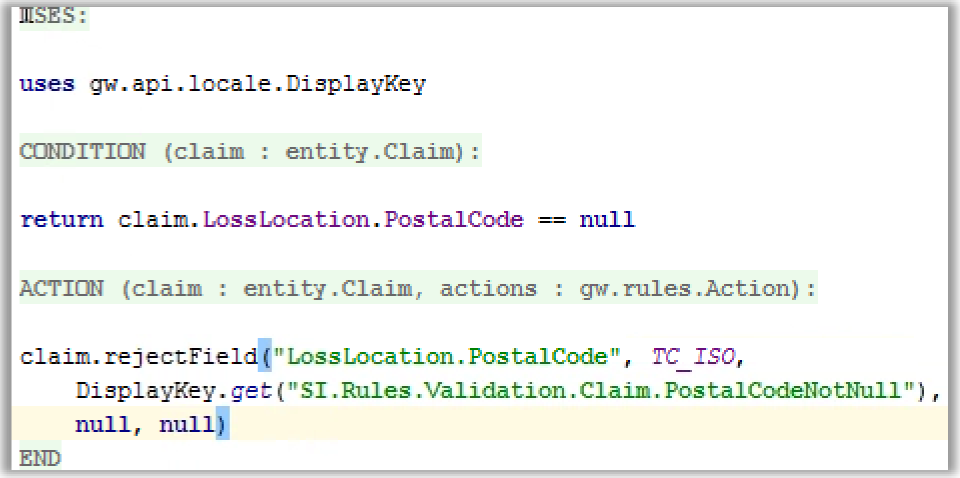
Claim Validation Rules
- ☑ CLV00100 - Theft and Police Reports
- ☑ CLV00200 - No-Fault Warning
- □ CLV01000 - Not set: Coverage in Question
- □ CLV02000 - Coverage indicated but no details prov
- ☑ CLV03000 - Future loss date
- ☑ CLV04000 - ClaimContact Role Configuration
- ☑ CLV05000 - Policy expiration date after effective d
- ☑ CLV06000 - Policy original effective date
- ☑ CLV06500 - Total liability percentage in evaluation
- ☑ CLV07000 - Liability can not exceed 100
- ☑ CLV08000 - Expected Recovery exceeds 100
- ☑ CLV09000 - ISO Validation
  - ☑ CLV09100 - Not Draft
  - ☑ CLV09200 - Policy
  - ☑ CLV09300 - LossLocation
  - ☑ CLV09400 - PolicyNumber
  - ☑ CLV09500 - PolicyType
  - ☑ CLV09600 - ClaimNumber
  - ☑ CLV09700 - LossDate
  - ☑ CLV09800 - State
  - ☑ CLV09850 - Null Postal Code
  - ☑ CLV09900 - Insured



```
USES:

uses gw.api.locale.DisplayKey

CONDITION (claim : entity.Claim):

return (claim.LossType == LossType.TC_AUTO) and
    (claim.LossCause == LossCause.TC_THEFTENTIRE or
     claim.LossCause == LossCause.TC_THEFTPARTS) and
    claim.MetroReports.length == 0

ACTION (claim : entity.Claim, actions : gw.rules.Action):

claim.reject(ValidationLevel.TC_NEWLOSS, DisplayKey.get("SI.Rules.Validation.Claim.PoliceReportNotNull"),
    null, null)
END
```



```
USES:

uses gw.api.locale.DisplayKey

CONDITION (claim : entity.Claim):
return (claim.FaultRating == FaultRating.TC_NOFAULT) and
    (claim.LossLocation.State == State.TC_AZ or
    claim.LossLocation.State == State.TC_CA)
ACTION (claim : entity.Claim, actions : gw.rules.Action):

claim.reject(null, null,
    ValidationLevel.TC_NEWLOSS,
    DisplayKey.get("SI.Rules.Validation.Claim.StateIsNotNoFault", claim.LossLocation.State))
END
```



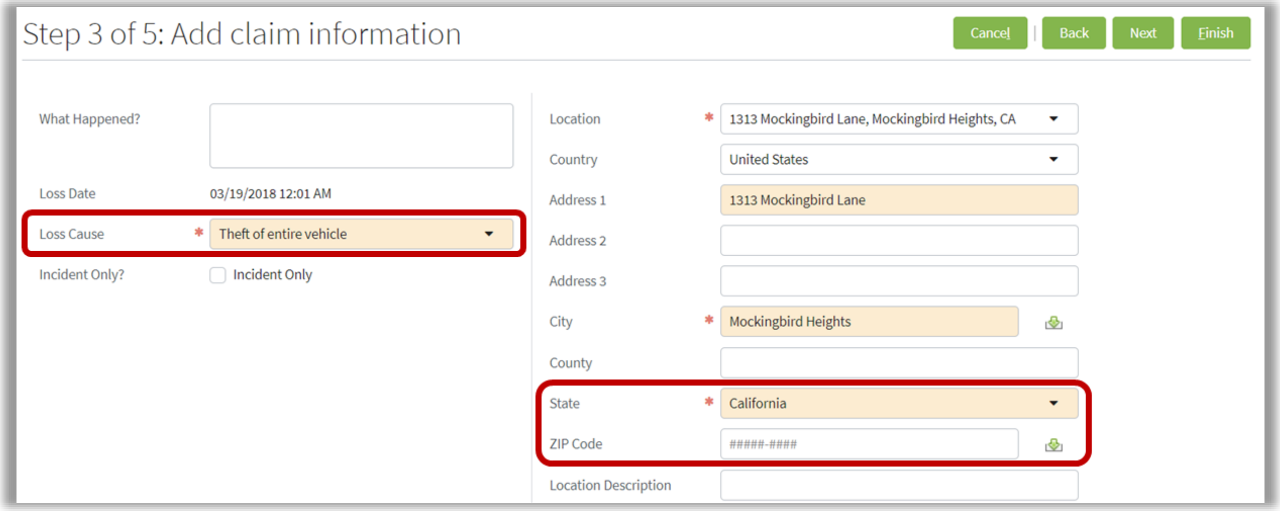
```
USES:

uses gw.api.locale.DisplayKey

CONDITION (claim : entity.Claim):

return claim.LossLocation.PostalCode == null

ACTION (claim : entity.Claim, actions : gw.rules.Action):

claim.rejectField("LossLocation.PostalCode", TC_ISO,
    DisplayKey.get("SI.Rules.Validation.Claim.PostalCodeNotNull"),
    null, null)
END
```

6.3.2 Test procedure

1. **Start ClaimCenter in debug mode.**
2. **Log in as aapplegate/gw.**
3. **Create a new personal auto claim.**

   On the Loss Details step, make sure that:

   - The Postal Code/ZIP Code is left blank.
   - The Loss Cause is set to one of the Theft options.
   - The Loss Location State is set to California or Arizona.
   - There are no Police Reports associated with the claim.
   - The Fault Rating is set to No Fault.



Step 3 of 5: Add claim information

What Happened?

Loss Date 03/19/2018 12:01 AM

Loss Cause * Theft of entire vehicle

Incident Only? □ Incident Only

Location * 1313 Mockingbird Lane, Mockingbird Heights, CA

Country United States

Address 1 1313 Mockingbird Lane

Address 2

Address 3

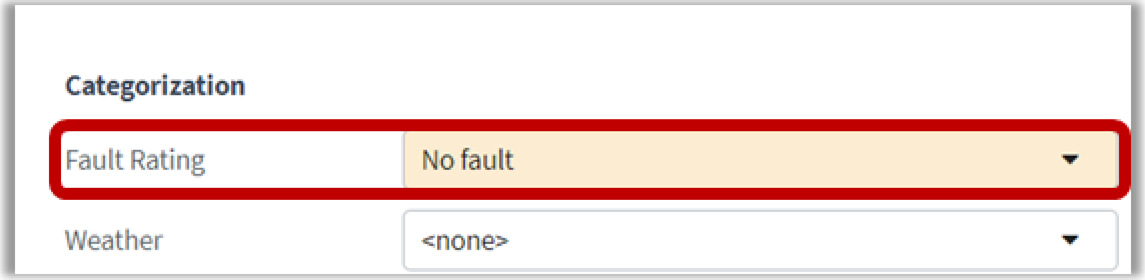
City * Mockingbird Heights

County

State * California

ZIP Code #####-####

Location Description

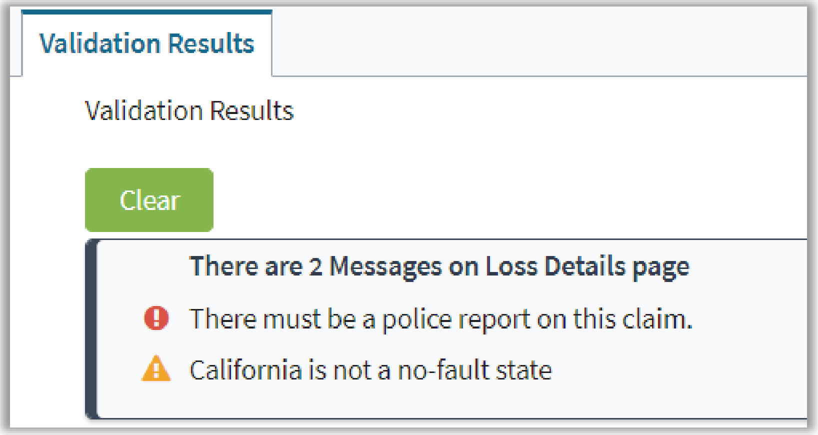


Categorization

Fault Rating No fault

Weather <none>

4. **Attempt to save the claim.**
   a) Verify that it cannot be saved due to the missing police report.
   b) Verify that the warning concerning No Fault appears.



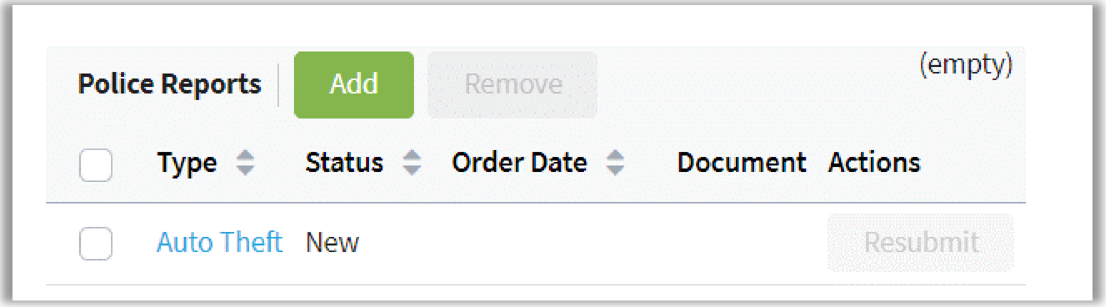
Validation Results

There are 2 Messages on Loss Details page

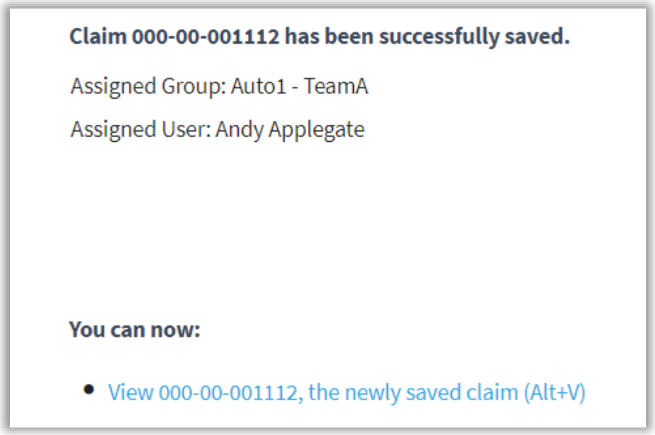
❗ There must be a police report on this claim.

⚠ California is not a no-fault state

5. **Add a police report and attempt to save the claim.**
   Verify that the claim validation level is New Loss Completion.



□	Type	Status	Order Date	Document	Actions
□	Auto Theft	New			Resubmit



Claim 000-00-001112 has been successfully saved.

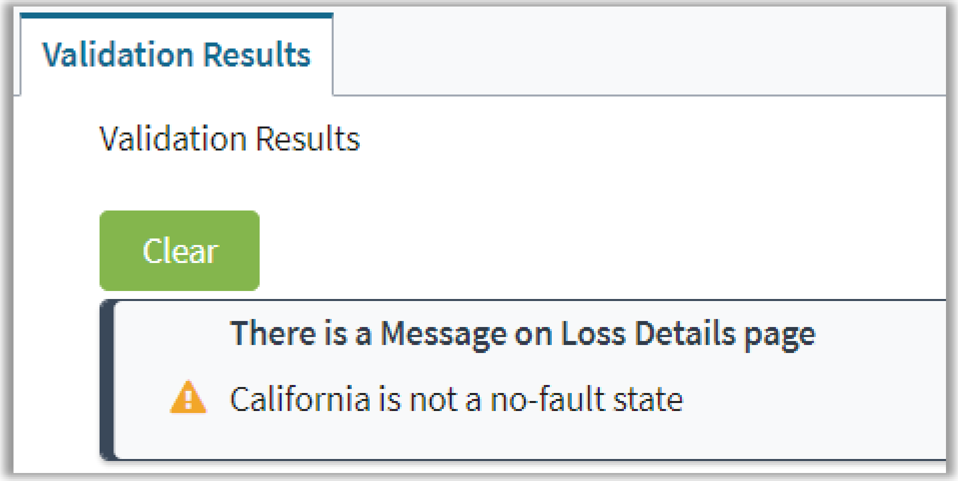
Assigned Group: Auto1 - TeamA

Assigned User: Andy Applegate

You can now:

- View 000-00-001112, the newly saved claim (Alt+V)

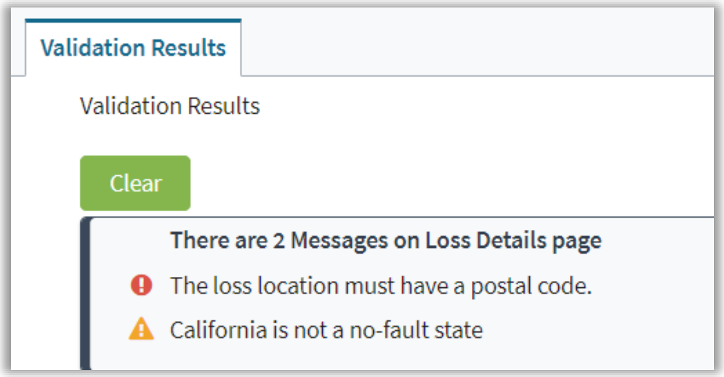
6. **Using Actions → Validate Claim Only, validate the claim at the New Loss Completion level.** Verify that the warning concerning No Fault appears.



There is a Message on Loss Details page

⚠ California is not a no-fault state

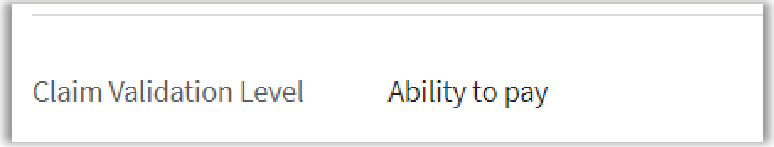
7. **Using Actions → Validate Claim Only, validate the claim at the Valid for ISO level.** Verify that an error is reported for the missing postal/ZIP code.



There are 2 Messages on Loss Details page

- The loss location must have a postal code.
- California is not a no-fault state

8. **Add a postal code to the loss location and save the claim.**
   Verify that the claim is saved at the Valid for ISO level or greater.



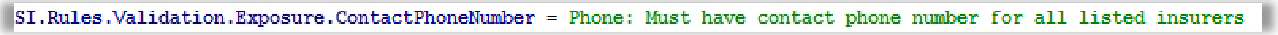
Claim Validation Level   Ability to pay

6.3.3 Validation for exposure lists

Solution

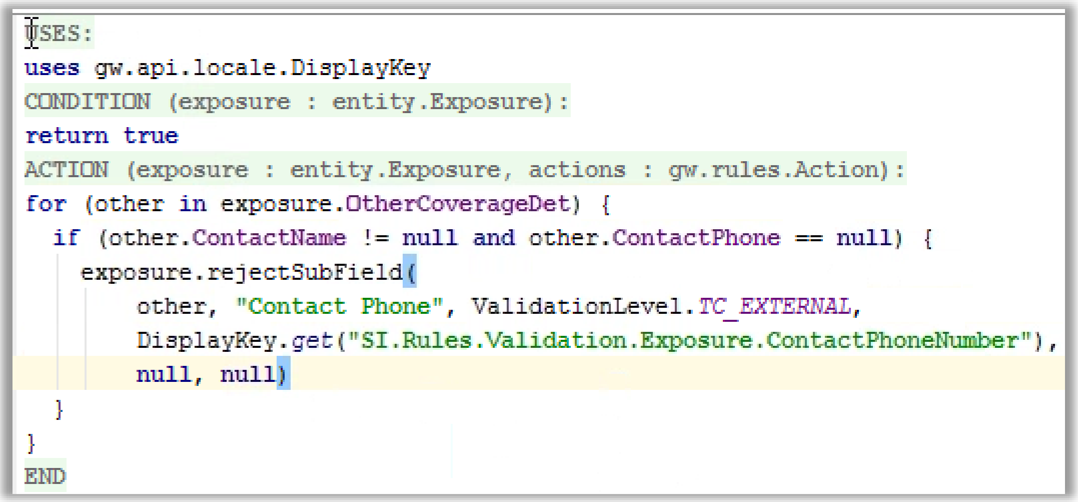
Exact details on how to complete the lab.

1. **Create a display key to report an error if an exposure has a listed insurer contact, but no phone number for that contact.**



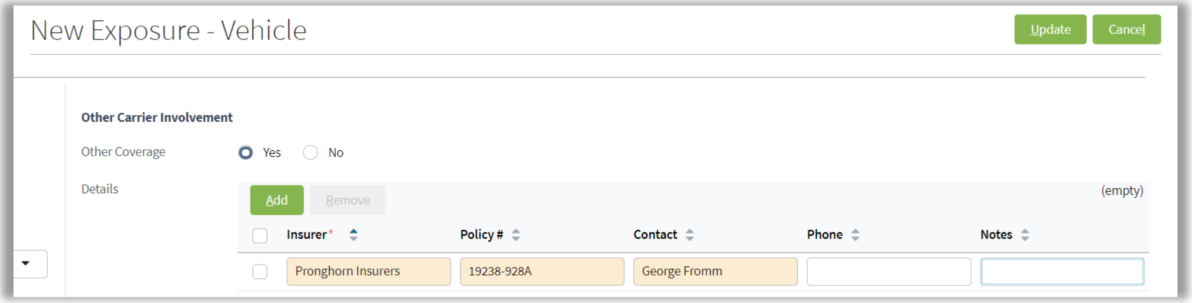
```
SI.Rules.Validation.Exposure.ContactPhoneNumber = Phone: Must have contact phone number for all listed insurers
```

2. **Create the validation rule to meet this requirement.**



```
USES:
uses gw.api.locale.DisplayKey
CONDITION (exposure : entity.Exposure):
return true
ACTION (exposure : entity.Exposure, actions : gw.rules.Action):
for (other in exposure.OtherCoverageDet) {
  if (other.ContactName != null and other.ContactPhone == null) {
    exposure.rejectSubField(
        other, "Contact Phone", ValidationLevel.TC_EXTERNAL,
        DisplayKey.get("SI.Rules.Validation.Exposure.ContactPhoneNumber"),
        null, null)
  }
}
END
```

3. **Test procedure:**
   Note: To simplify testing, disable the Exposure ISO Validation Rules (EXV05000)
   a) Attempt to create a new collision exposure where there is at least one Other Carrier Involvement with a contact and no phone number.



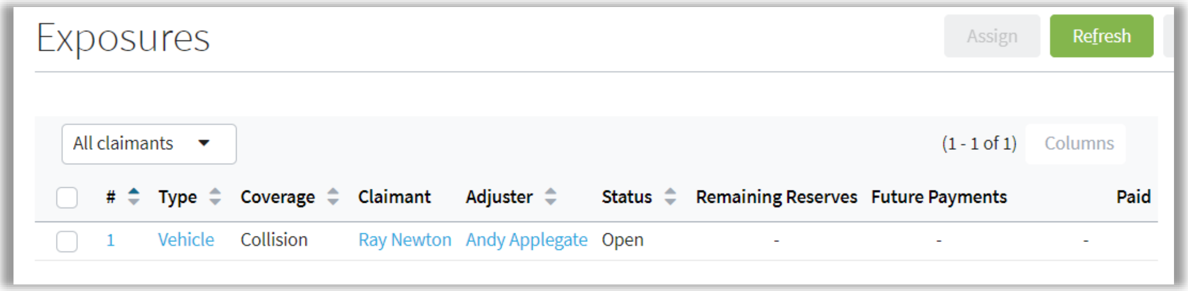
Other Carrier Involvement

Other Coverage ● Yes ○ No

Details (empty)

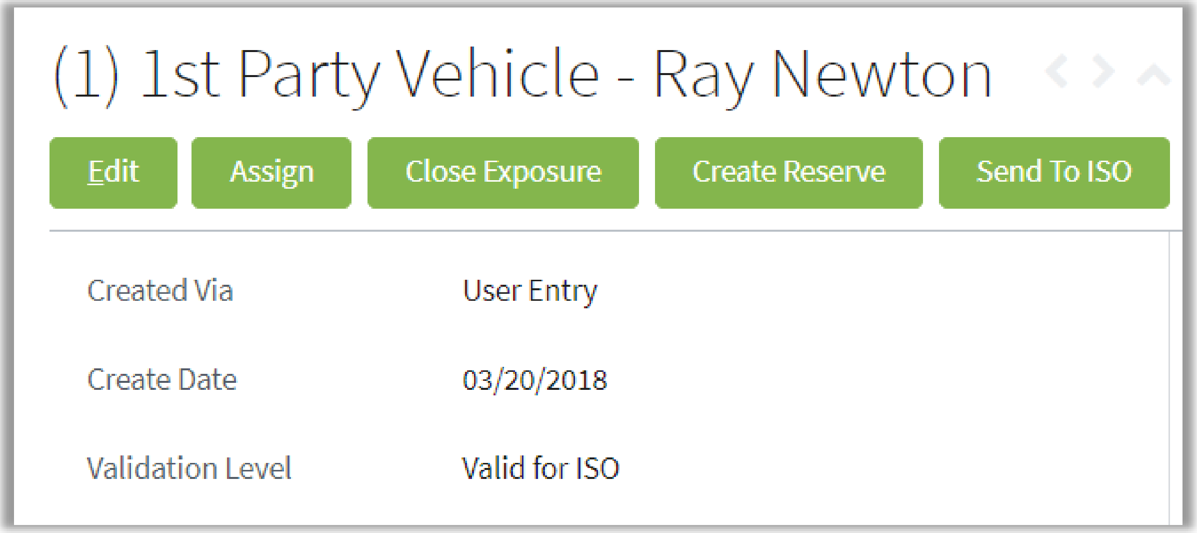
□	Insurer*	Policy #	Contact	Phone	Notes
□	Pronghorn Insurers	19238-928A	George Fromm		

   b) Verify that the exposure can be saved, but that it cannot reach the Send to external validation level.

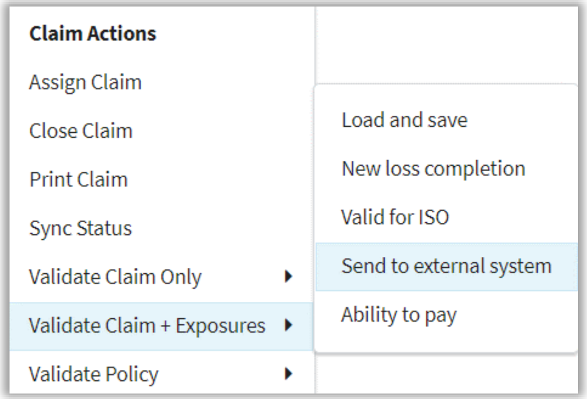


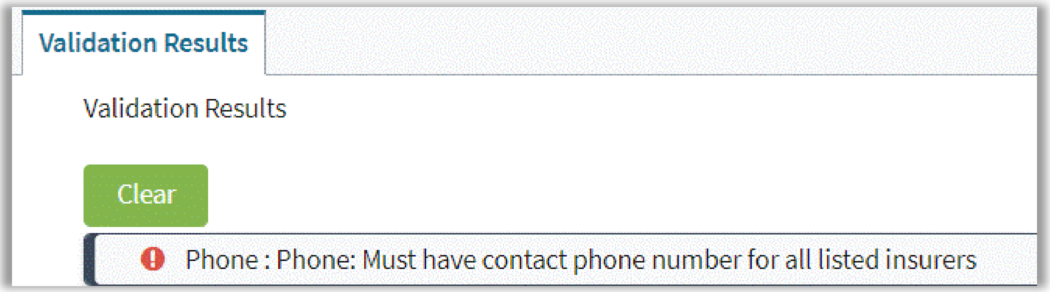
(1 - 1 of 1)

□	#	Type	Coverage	Claimant	Adjuster	Status	Remaining Reserves	Future Payments	Paid
□	1	Vehicle	Collision	Ray Newton	Andy Applegate	Open	-	-	-



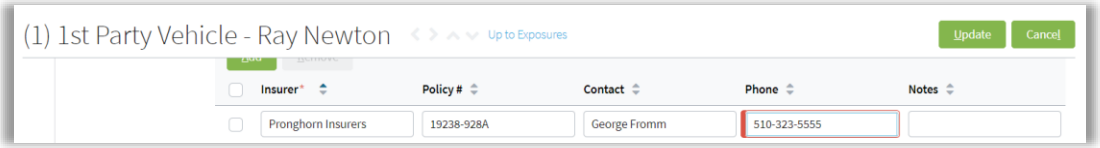
Created Via	User Entry
Create Date	03/20/2018
Validation Level	Valid for ISO





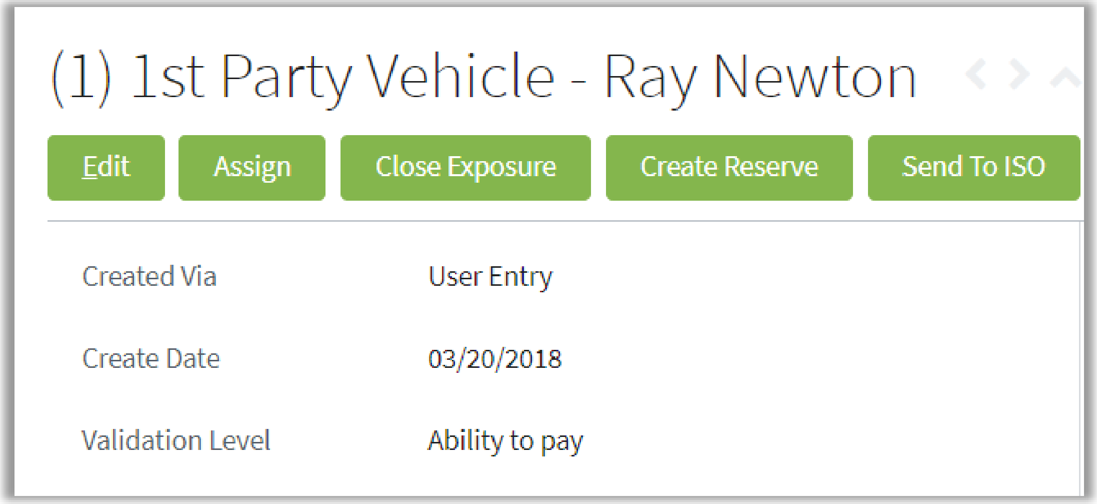
Phone : Phone: Must have contact phone number for all listed insurers

c) Add phone numbers to the carriers with contacts.



Insurer*	Policy #	Contact	Phone	Notes
Pronghorn Insurers	19238-928A	George Fromm	510-323-5555	

d) Verify that the exposure is now at or beyond the Send to external validation level.

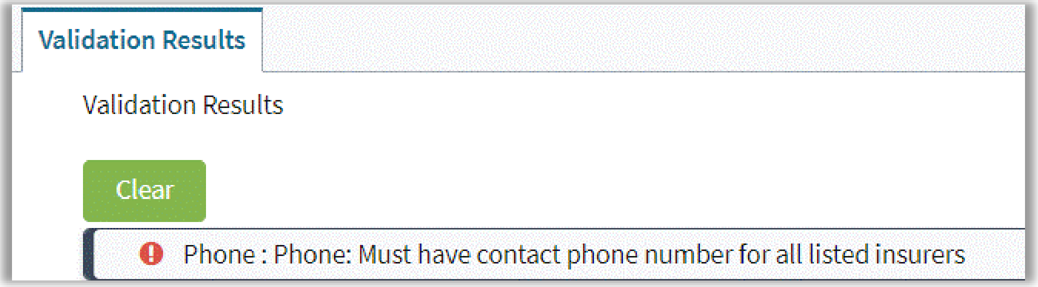


Created Via User Entry

Create Date 03/20/2018

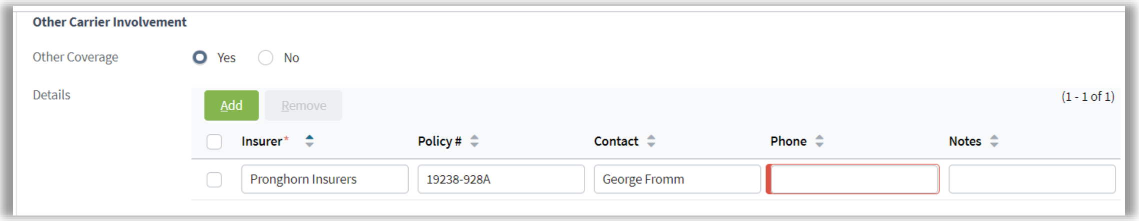
Validation Level Ability to pay

e) Attempt to remove the phone number from a carrier with a contact, and verify that the change cannot be saved and that the field is flagged.



Validation Results

Phone : Phone: Must have contact phone number for all listed insurers



Other Carrier Involvement

Other Coverage ● Yes ○ No

Details

(1 - 1 of 1)

□	Insurer*	Policy #	Contact	Phone	Notes
□	Pronghorn Insurers	19238-928A	George Fromm		

Lesson 7

Claim Setup Rules

When a claim is created in the New Claim Wizard, or is entered through the portal, or is imported from an external FNOL application, it enters the automated claim setup process. This process consists of several Gosu rule sets, which perform a number of actions. However, their primary purpose is to segment and assign the claim, and to create an initial workplan of activities for it.

7.1 Exploration

7.1.1 Investigation

1. **In the base configuration of ClaimCenter, under what conditions is a Property exposure's segment set to *prop_low*?**

2. **In the base configuration of ClaimCenter, what segment is a property exposure assigned to if the conditions for the Property Low, Property Medium, and Property High rules are all false?**

7.2 Set note for unverified policies

Under certain circumstances, Succeed Insurance adjusters and service representatives may enter a claim into ClaimCenter with an unverified policy. When this occurs, Succeed wants a note added to the claim reminding the adjuster to verify the policy before any payments are made.

Prerequisites

This lesson assumes that you have successfully completed Lesson 2, Writing Gosu Rules.

For this lab, you will use ClaimCenter, Guidewire Studio, and a supported web browser.

Studio should be running when you start the lab.

ClaimCenter Server should be running in debug mode when you start the lab.

The default URL for ClaimCenter is `http://localhost:8080/cc/ClaimCenter.do`.

7.2.1 Lab

Activity

1. **Which rule set can be used to accomplish this?**

2. **Add entries to display.properties for the subject and body of such a note.**
3. **Code a rule that creates the note.**
   The easiest way to accomplish this is to create the note and then attach it to the claim using the setClaim() method of the Note entity.

7.2.2 Test procedure

1. **Reload changed classes.**
2. **Switch to ClaimCenter and log in as aapplegate/gw.**
3. **Create a new General Liability claim using an unverified policy.**
4. **Navigate to the new claim.**
5. **Verify that the Unverified Policy note has been added to the claim.**

7.3 Segmentation for property claim fraud detection

7.3.1 Lab

Succeed Insurance has experienced a rise in fraudulent property claims where a loss occurred. For example, a new policy was issued, and a loss was reported shortly after the policy came into effect. Succeed wants to use claim segmentation to help identify claims which are possibly fraudulent.

Activity

Tip

Modifying a typelist constitutes a change to the data model. When you do this, the project must be rebuilt(**Build ➔ Build Project** or **Ctrl+F9**) for the new typecodes to be available for code.

1. **Modify the Claim Segment typelist by adding a typecode named “Early Term Loss”, which will be visible in the UI.**
2. **Modify the claim segmentation rules so that if a property claim has a loss date which is within 30 days of the policy's effective date, the claim is segmented as an “Early Term Loss”.**
   a) This rule must take precedence over all other property segmentation rules. Give the rule a name that is descriptive of its purpose and conforms to best practices for naming rule sets.
   b) The segmentation rule will need a date library function.
   c) If your segmentation rule segments the claim, then your code should exit the rule set.

7.3.2 Test procedure

1. **Stop the ClaimCenter server and restart it in debug mode.**
2. **Log on as Ronald Barnes (rbarnes/gw).**
3. **Using policy number 23-502011, create a property claim.**
   a) Set the Loss Date to less than ten days after the Policy Effective Date.
   b) Complete the claim, using defaults where possible and making up data where necessary.
   c) Assign the claim to Roland Barnes.

4. **Verify that the claim is segmented as “Early term loss”.**

7.4 Solutions

Solution

Exact details on how to complete the lab.

7.4.1 Investigation

1. In the base configuration of ClaimCenter, under what conditions is a Property exposure's segment set to *prop_low*?

The loss type is property damage, AND at least one of the following things is true:

- *The severity of the Exposure's Incident is medium;*
- *The Exposure Type is Loss of Use due to damage;*
- *The Exposure is a loss to a Third Party.*

2. In the base configuration of ClaimCenter, what segment is a property exposure assigned to if the conditions for the Property Low, Property Medium, and Property High rules are all false?

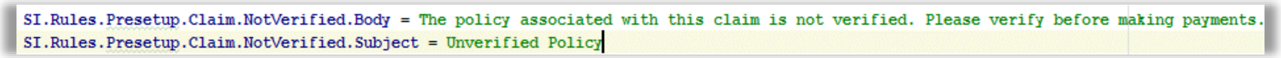
The prop_mid segment, which is accomplished by the Property Default rule.

7.4.2 Lab

1. Which rule set can be used to accomplish this?

Claim Presetup

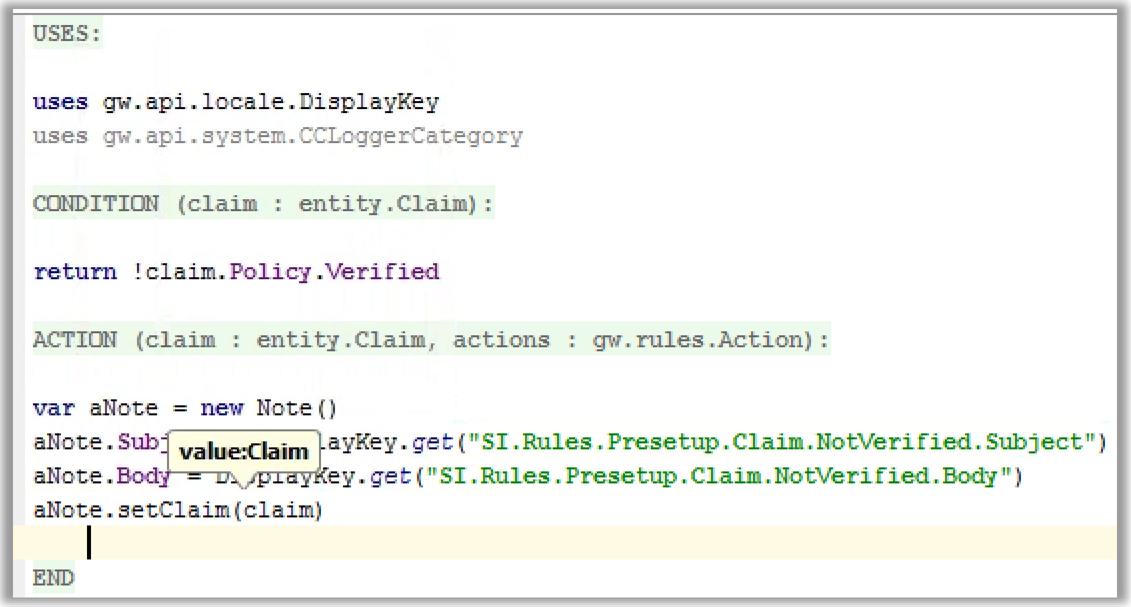
2. Add entries to display.properties for the subject and body of such a note.



```
SI.Rules.Presetup.Claim.NotVerified.Body = The policy associated with this claim is not verified. Please verify before making payments.
SI.Rules.Presetup.Claim.NotVerified.Subject = Unverified Policy
```

3. Code a rule that creates the note.

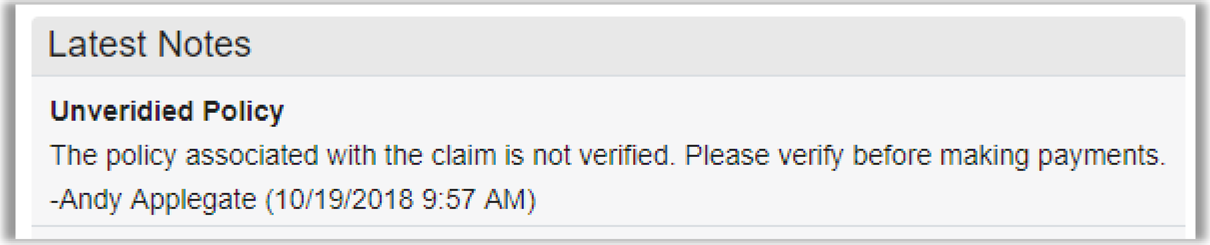
The easiest way to accomplish this is to create the note and then attach it to the claim using the setClaim() method of the Note entity.



```
USES:

uses gw.api.locale.DisplayKey
uses gw.api.system.CCLoggerCategory

CONDITION (claim : entity.Claim):

return !claim.Policy.Verified

ACTION (claim : entity.Claim, actions : gw.rules.Action):

var aNote = new Note()
aNote.Subj ayKey.get("SI.Rules.Presetup.Claim.NotVerified.Subject")
aNote.Body = D playKey.get("SI.Rules.Presetup.Claim.NotVerified.Body")
aNote.setClaim(claim)

END
```

7.4.3 Test procedure

1. **Reload changed classes.**
2. **Switch to ClaimCenter and log in as aapplegate/gw.**
3. **Create a new General Liability claim using an unverified policy.**
4. **Navigate to the new claim.**
5. **Verify that the Unverified Policy note has been added to the claim.**



Latest Notes

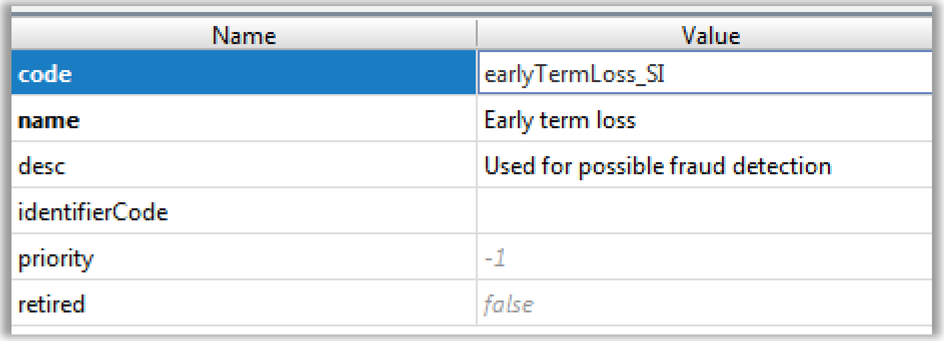
Unveridied Policy

The policy associated with the claim is not verified. Please verify before making payments.

-Andy Applegate (10/19/2018 9:57 AM)

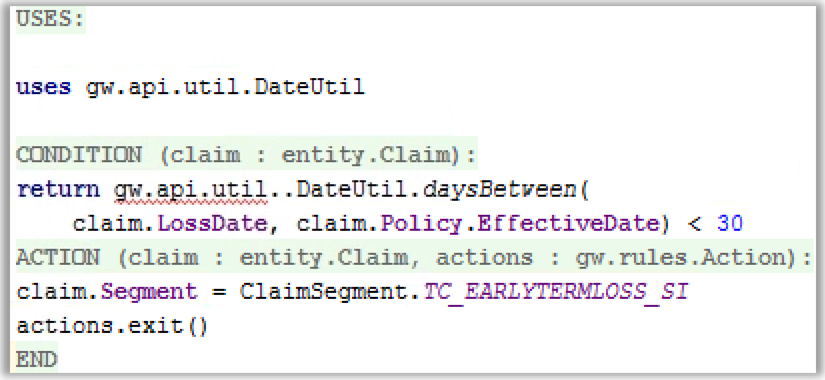
7.4.4 Lab

1. **Modify the Claim Segment typelist by adding a typecode named “Early Term Loss”, which will be visible in the UI.**



Name	Value
code	earlyTermLoss_SI
name	Early term loss
desc	Used for possible fraud detection
identifierCode	
priority	-1
retired	false

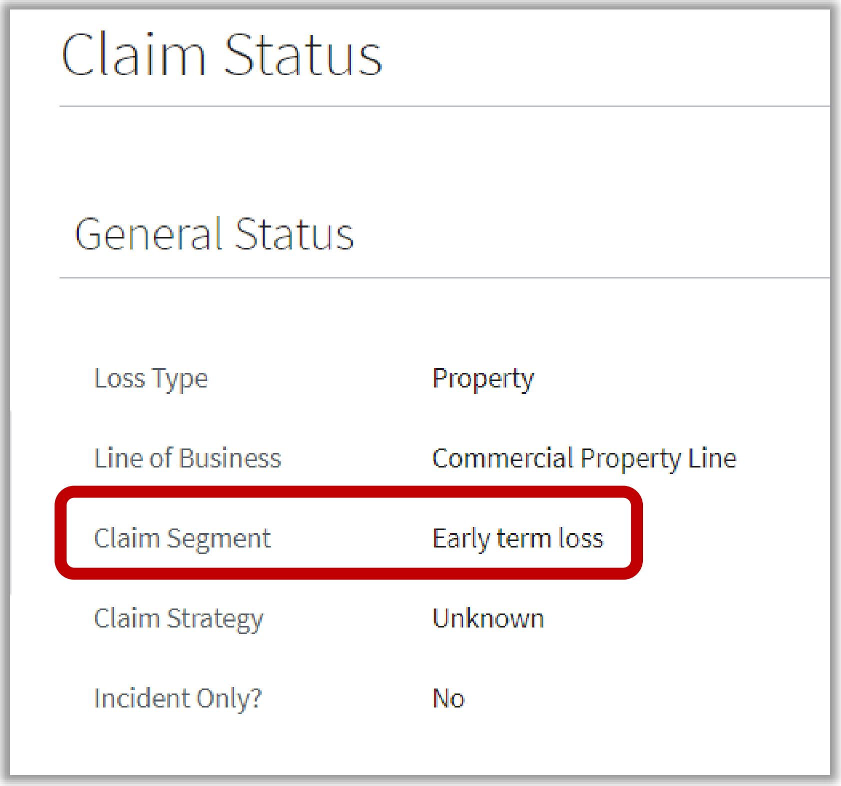
2. **Modify the claim segmentation rules so that if a property claim has a loss date which is within 30 days of the policy's effective date, the claim is segmented as an "Early Term Loss".**
   a) This rule must take precedence over all other property segmentation rules. Give the rule a name that is descriptive of its purpose and conforms to best practices for naming rule sets.
   b) The segmentation rule will need a date library function.
   c) If your segmentation rule segments the claim, then your code should exit the rule set.



```
USES:

uses gw.api.util.DateUtil

CONDITION (claim : entity.Claim):
return gw.api.util..DateUtil.daysBetween(
    claim.LossDate, claim.Policy.EffectiveDate) < 30
ACTION (claim : entity.Claim, actions : gw.rules.Action):
claim.Segment = ClaimSegment.TC_EARLYTERMLOSS_SI
actions.exit()
END
```

7.4.5 Test procedure

1. **Start the ClaimCenter server in debug mode.**
2. **Log on as Ronald Barnes (rbarnes/gw).**
3. **Using policy number 23-502011, create a property claim.**
   a) Set the Loss Date to less than ten days after the Policy Effective Date.
   b) Complete the claim, using defaults where possible and making up data where necessary.
   c) Assign the claim to Roland Barnes.
4. **Verify that the claim is segmented as "Early term loss".**



Claim Status

General Status

Loss Type	Property
Line of Business	Commercial Property Line
Claim Segment	Early term loss
Claim Strategy	Unknown
Incident Only?	No

Lesson 8

Configuring Claim Contacts

A given claim may have many associated contacts (persons and companies).

ClaimCenter has a rich contact model that helps agents and adjusters keep track of the roles each contact plays with respect to a given claim.

8.1 Adding a claim contact role

Prerequisites

This lesson assumes that you have successfully completed Lesson 2, Writing Gosu Rules.

For this lab, you will use ClaimCenter, Guidewire Studio, and a supported web browser.

Studio should be running when you start the lab.

ClaimCenter Server should be running in debug when you start the lab.

ContactManager Server should be running when you start the lab.

The default URL for ClaimCenter is **`http://localhost:8080/cc/ClaimCenter.do`**.

You should be logged in to ClaimCenter as **`su/gw`**.

8.1.1 Lab

Some of Succeed Insurance's customers are not fluent in English. During claim negotiations, Succeed provides the services of an interpreter as needed.

The Interpreter's name should be displayed in the Negotiation Details, just below the Negotiation Contact.

Activity

1. **Determine the name of the detail view where the interpreter field should appear.**

2. **Create a display key that displays the word "Interpreter".**
3. **Add typecodes to the ContactRole and ContactRoleCategory typelists to implement the role.**

Remember to use the "_SI" suffix

4. **Add the typecode to entityconstraints-config.xml.**
   Interpreters are persons and vendors, associated only with Negotiations. There can only be one interpreter per negotiation.
5. **In Studio, open the detail view discovered in Step 1.**
6. **Create a Claim Contact Input to display the negotiator's name.**

8.1.2 Test procedure

1. **Start (or restart) the ClaimCenter server.**
   a) If necessary, log out of ClaimCenter first.
   b) When the server has started, log in as **aapplegate/gw**.
2. **Create a negotiation.**
   a) Navigate to claim **235-53-365870** (the Ray Newton claim).
   b) Select Plan of Action ➔ Negotiations
   c) Click New Negotiation.
   d) In the **Name** field, enter **Test Negotiation**.
3. **Verify that you can add an Interpreter.**
   a) Select Parties Involved ➔ Contacts.
   b) Select **New Contact ➔ Vendor ➔ Doctor**.
   (Note: In a full implementation, an Interpreter would be assigned to a new category under Vendor. Since the Interpreter must be a PersonVendor, you must select Doctor.)
   c) In the **Last Name** field, enter **Bancroft**.
   d) Provide Bancroft with a legitimate **TaxID**.
   e) In the **Roles** list, click **Add**.
   f) Try to add the Interpreter role. Can you?

   g) In the Related To column, select Test Negotiation.
   h) Can you add the Interpreter role now?

   i) Click **Update**.
4. **Verify that there can only be one Interpreter on a Negotiation.**
   a) Select New Contact ➔ Vendor ➔ Doctor.
   b) In the **Last Name** field, enter **Duke**.
   c) In the **Roles** list, click **Add**.
   d) In the Related To column, select Test Negotiation.
   e) Can you add the Interpreter role?

f) Click **Update**. What happens?

g) Click **Cancel**.

5. Verify the claim contact widget.

a) Select Plan of Action ➔ Negotiations.

b) Click Test Negotiation.

c) Does the Interpreter field appear?

d) Does Bancroft appear in the Interpreter field?

8.2 Solution

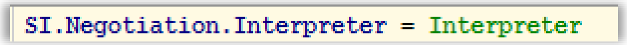
8.2.1 Lab

Solution

Exact details on how to complete the lab.

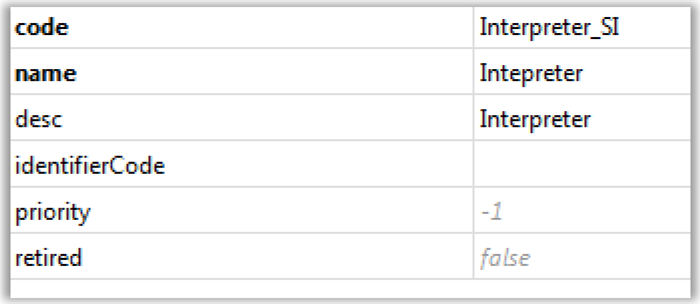
1. **Determine the name of the detail view where the interpreter field should appear.** ***ClaimNegotiationDetailsDV.pcf (open the view in ClaimCenter and click Alt+Shift+I to open the Location Info window).***

2. **Create a display key that displays the word "Interpreter".**

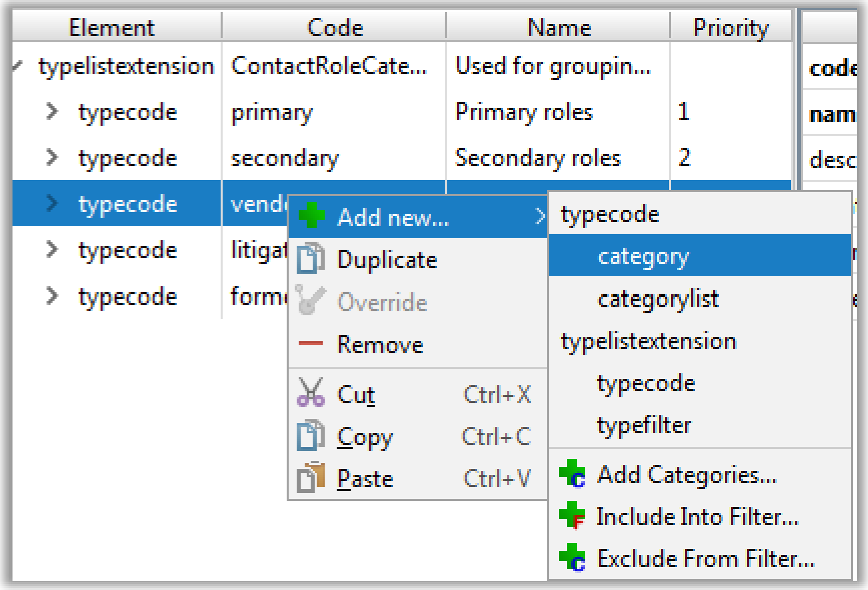


```
SI.Negotiation.Interpreter = Interpreter
```

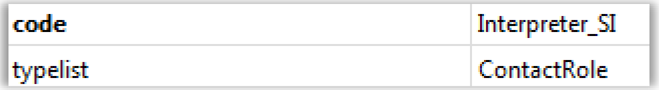
3. **Add typecodes to the ContactRole and ContactRoleCategory typelists to implement the role.** Remember to use the "_SI" suffix



code	Interpreter_SI
name	Intepreter
desc	Interpreter
identifierCode	
priority	*-1*
retired	*false*

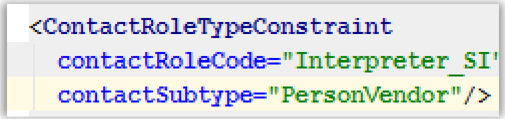


Element	Code	Name	Priority
typelistextension	ContactRoleCate...	Used for groupin...	
typecode	primary	Primary roles	1
typecode	secondary	Secondary roles	2
typecode	vend		
typecode	litigat		
typecode	form		

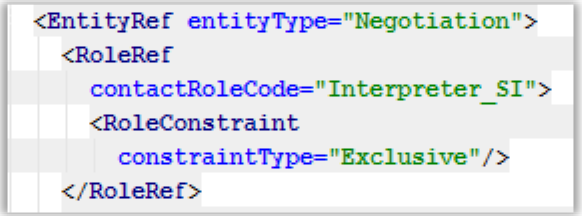


code	Interpreter_SI
typelist	ContactRole

4. **Add the typecode to entityconstraints-config.xml.**
   Interpreters are persons and vendors, associated only with Negotiations. There can only be one interpreter per negotiation.

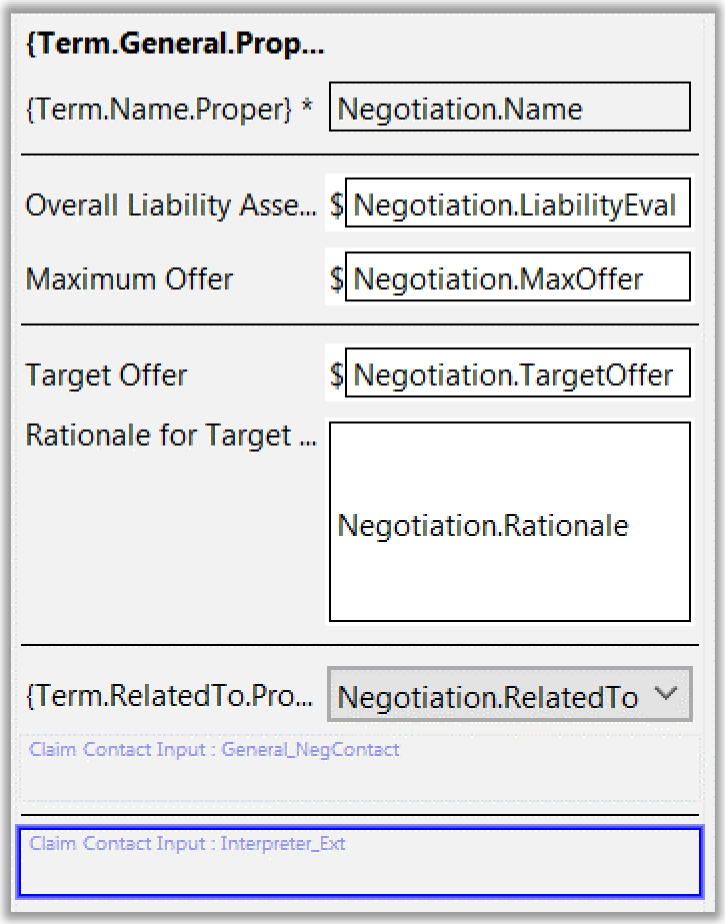


```
<ContactRoleTypeConstraint
  contactRoleCode="Interpreter_SI'
  contactSubtype="PersonVendor"/>
```



```
<EntityRef entityType="Negotiation">
  <RoleRef
    contactRoleCode="Interpreter_SI">
    <RoleConstraint
      constraintType="Exclusive"/>
  </RoleRef>
```

5. **In Studio, open the detail view you discovered in Step 1.**
6. **Create a Claim Contact Input to display the negotiator’s name.**



{Term.General.Prop...

{Term.Name.Proper} * Negotiation.Name

Overall Liability Asse... $ Negotiation.LiabilityEval

Maximum Offer $ Negotiation.MaxOffer

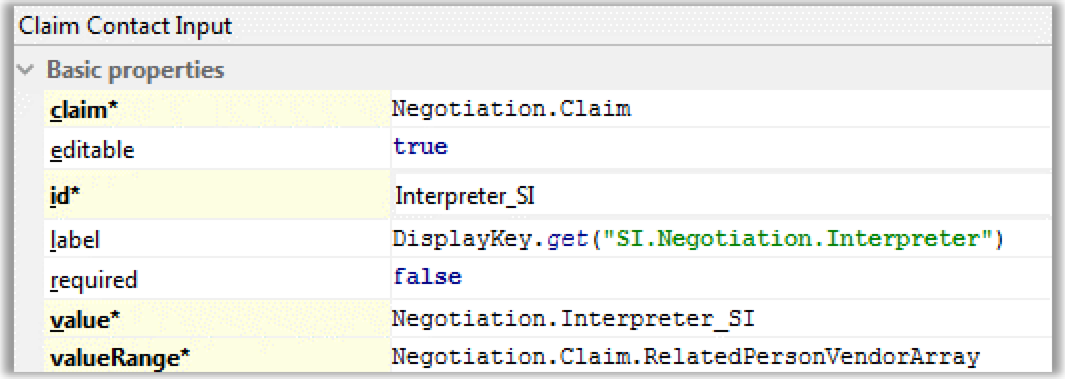
Target Offer $ Negotiation.TargetOffer

Rationale for Target ... Negotiation.Rationale

{Term.RelatedTo.Pro... Negotiation.RelatedTo

Claim Contact Input : General_NegContact

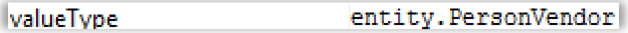
Claim Contact Input : Interpreter_Ext



Claim Contact Input

Basic properties

claim*	Negotiation.Claim
editable	true
id*	Interpreter_SI
label	DisplayKey.get("SI.Negotiation.Interpreter")
required	false
value*	Negotiation.Interpreter_SI
valueRange*	Negotiation.Claim.RelatedPersonVendorArray

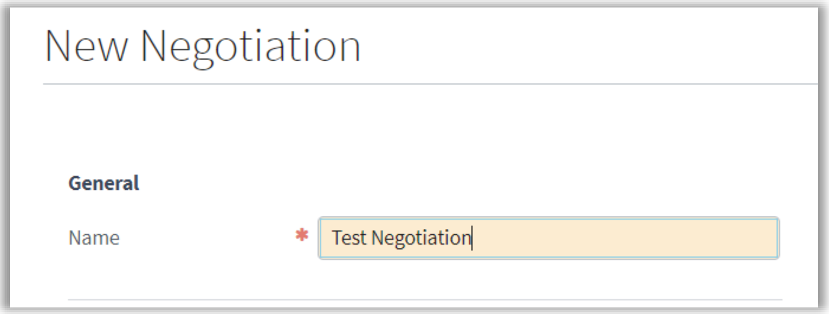


valueType	entity.PersonVendor

8.2.2 Test procedure

1. **Start (or restart) the ClaimCenter server.**
   a) If necessary, log out of ClaimCenter first.
   b) When the server has started, log in as **aapplegate/gw**.

2. **Create a negotiation.**
   a) Navigate to claim **235-53-365870** (the Ray Newton claim).
   b) Select Plan of Action ➔ Negotiations
   c) Click New Negotiation.
   d) In the **Name** field, enter **Test Negotiation**.



New Negotiation

General

Name * Test Negotiation

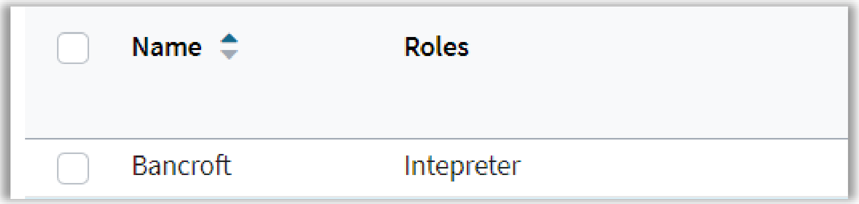
   e) Click **Update**.

3. **Verify that you can add an Interpreter.**
   a) Select Parties Involved ➔ Contacts.
   b) Select **New Contact ➔ Vendor ➔ Doctor**.
   (Note: In a full implementation, an Interpreter would be assigned to a new category under Vendor. Because Interpreter must be a PersonVendor, you must select Doctor.)
   c) In the **Last Name** field, enter **Bancroft**.
   d) Provide Bancroft with a legitimate **TaxID**.

e) In the Roles list, click **Add**. Try to add the Interpreter role. Can you?
No

f) In the Related To column, select Test Negotiation.
g) Can you add the Interpreter role now?
Yes

h) Click **Update**.



□	Name	Roles
□	Bancroft	Intepreter

4. **Verify that there can only be one Interpreter on a Negotiation.**
   a) Select New Contact ➔ Vendor ➔ Doctor.
   b) In the **Last Name** field, enter **Duke**.
   c) In the **Roles** list, click **Add**.
   d) In the Related To column, select Test Negotiation.
   e) Can you add the Interpreter role?
   Yes

   f) Click **Update**. What happens?
   An error message is displayed saying that you cannot add Interpreter to this Contact because another Contact already has that role.
   g) Click **Cancel**.

5. **Verify the claim contact widget.**
   a) Select Plan of Action ➔ Negotiations.
   b) Click Test Negotiation.
   c) Does the Interpreter field appear?
   Yes

   d) Does Bancroft appear in the Interpreter field?
   Yes