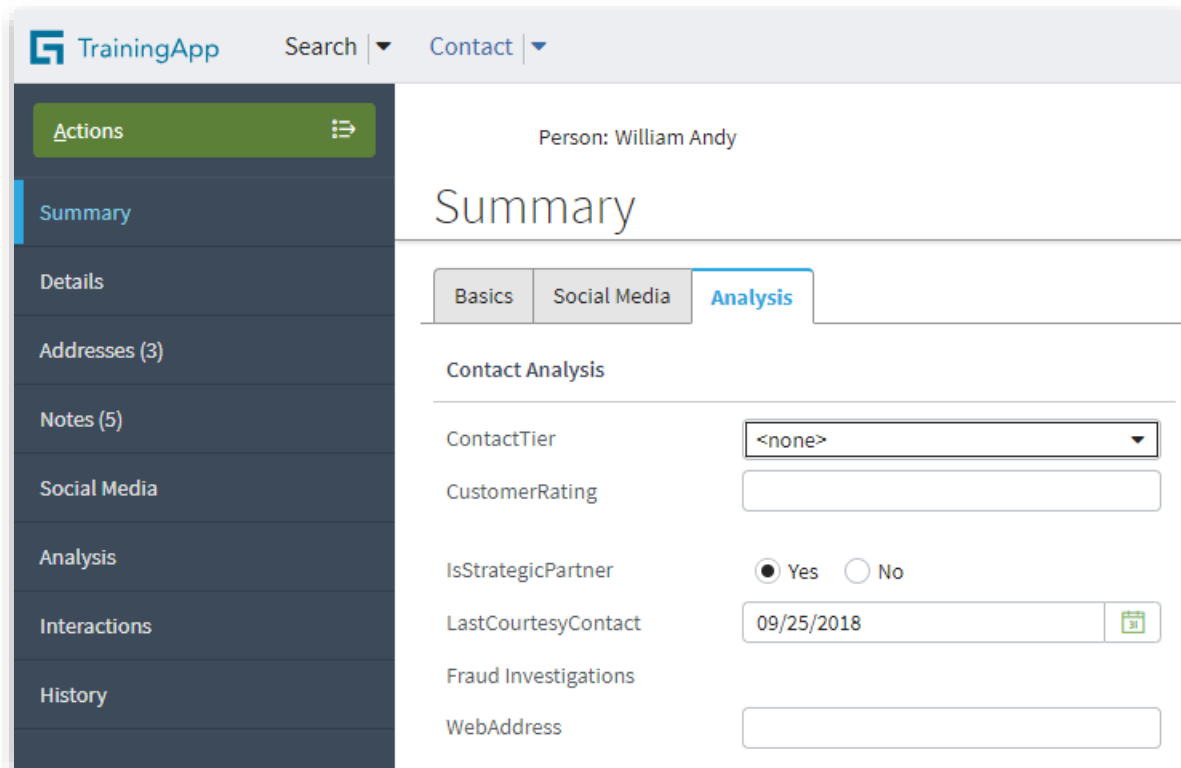


## Lesson 5 Atomic widgets

An insurance company wants to extend TrainingApp functionality by capturing more details about each contact. The complete requirement will be implemented over multiple modules. Recall that you have already implemented the required **data model changes** in the previous *Extending Base Application Entities* lesson.



TrainingApp Search Contact

Person: William Andy

### Summary

Basics Social Media Analysis

#### Contact Analysis

ContactTier	<none>
CustomerRating	
IsStrategicPartner	<input checked="" type="radio"/> Yes <input type="radio"/> No
LastCourtesyContact	09/25/2018
Fraud Investigations	<input type="checkbox"/>
WebAddress	

In this exercise your job is to make the necessary **user interface changes** to meet customer requirements. As a configuration developer you will use the PCF Editor in Guidewire Studio to modify the TrainingApp UI.

In this module, you will first practice the DOT notation. Next, you will review the requirements. Finally, you will configure the appropriate PCF file to display the fields and deploy your changes.

### 5.1 Prerequisites

You must first complete the following previous module(s):

#### 1. Extending Base Application Entities

For this exercise, use TrainingApp, Guidewire Studio, and a supported web browser.

<http://localhost:8880/ab/ContactManager.do> is the default URL for TrainingApp. To view, edit, and

delete various contacts, log in to TrainingApp as Alice Applegate. The login/password for Alice Applegate is aapplegate/gw.

## 5.2 Lab: Practice the Dot notation

As a configuration developer, you need to understand how to work with the dot notation. In this exercise, you will use the data dictionary to determine how to reference entities, fields, and properties using dot notation.

### 1. Open the data dictionary

- a) Open the data dictionary in your browser.

### 5.2.1 Write it down

For an object named "anABContact" that is of the type ABContact, identify the correct dot notation syntax for the following:

#### 1. The creation date and time for anABContact.

#### 2. The tax status for anABContact.

#### 3. The city for the primary address of anABContact.

#### 4. All the buildings associated with anABContact.

#### 5. Assuming that anABContact is a company, the number of employees for anABContact.

6. Assuming that anABContact is a legal venue, the type of court for anABContact.



## 5.3 Lab: Determining Widgets and their requirements

In this exercise, you will create new atomic widgets to capture and display additional contact details for an existing details view. Review the widget requirements received from the business analysts. This table summarizes the requirements for each widget. Refer to the wireframe at the beginning of the exercise to see how the fields will be displayed in the UI.

As a configuration developer, you want to be able to find fields in the data dictionary and determine their data model type. As a configuration developer, you want to be able to determine the appropriate widget and Gosu data type based on the data model type of a field.

### 1. Open the Data Dictionary

- Open the data dictionary in your browser.

### 2. Review the Widget reference table

Data type (Data Model)	Recommended widget(s) in a Detail View	Recommended widget(s) in a List View	Data type (Gosu)
varchar	Text Input	Text Cell	java.lang.String
integer	Text Input	Text Cell	java.lang.Integer
decimal	Text Input	Text Cell	java.math.BigDecimal
text	Text Input Text Area Input	Text Cell Text Area Cell	java.lang.String
date, datetime	Date Input	Date Cell	java.util.Date

Data type (Data Model)	Recommended widget(s) in a Detail View	Recommended widget(s) in a List View	Data type (Gosu)
bit	Boolean Dropdown Input Boolean Radio Button Input Check Box Input	Boolean Radio Cell Checkbox Cell	java.lang.Boolean
foreignkey	Range Input Range Radio Button Input	Range Cell	OtherEntityType
typekey	TypeKey Input TypeKey Radio Button Input	Typekey Cell	typekey.TypeListName

Table 1 Widget reference table

3. Use the Data Dictionary and the Widget reference table to complete the following table Note: All the fields are defined on ABContact entity

	Label/Info	Editable	Field	Datatype (Data Model)	Widget	valueType (Gosu)
1	Contact Analysis	n/a	n/a	n/a		n/a
2	We need a horizontal divider	n/a	n/a	n/a		n/a
3	Contact Tier	Yes				
4	Customer Rating	Yes				
5	Strategic Partner	Yes				

	Label/Info	Editable	Field	Datatype (Data Model)	Widget	valueType (Gosu)
6	Last Courtesy Contact	Yes				
7	Fraud Investigation Number	No				
8	Web Address	Yes				



Stop

## 5.4 Lab: Add new Atomic Widgets

As a configuration developer, you want to create atomic widgets; bind some of those atomic widgets to entity fields to display data from the database; and define user-friendly labels.

### 1. Open Guidewire Studio

- From the TrainingApp folder, open a command window and start Guidewire Studio for TrainingApp.
- Alternatively, use the Start TrainingApp Studio shortcut.

### 2. Run or Debug the TrainingApp project

- Use the main menu, toolbar, or keystroke to run or Debug the server.
- Verify that the Run or Debug Console tab shows \*\*\*\*\* ContactManager ready \*\*\*\*\*.

### 3. Log in to TrainingApp

- Log in as Alice Applegate.

### 4. View the William Andy Summary

- Search for William Andy.

- b) In the search results list view, navigate to the William Andy contact.

## 5. Open the Summary Page in Guidewire Studio

- a) Use the appropriate internal tools shortcut to open the PCF file in Guidewire Studio.



### Keyboard shortcut

Open a PCF file in Studio from the Browser

If you are running an open application project in Guidewire Studio and if internal tools are enabled, you can automatically open the location being viewed in the application. Use the `ALT + SHIFT + E` keystroke in the browser.

## 6. Open the ABContactSummaryCV

- a) In the PCF Editor, open the ABContactSummaryCV.pcf file.

## 7. Add widgets to the Detail View Panel on the Analysis card

- a) Use the Toolbox to add widgets.
- b) Use the Properties window to define the necessary widget properties when such as `id*`, `editable`, and `value*`.
- c) When defining labels, create display keys.
- d) Use your completed table and the wireframe as guidelines



### Best practices

Adhere to Best practices

Guidewire recommends PCF file should not contain "hard coded" display text. Benefits of display keys:

- Easily localize text and accommodate users from various locales
- Allows for labels with dynamic content
- Reusability – you must make modifications only at one place even if you decide to change a certain text that is shown at multiple places in the User Interface

The recommended naming convention for customer display keys, is to use a standard prefix such as "Ext". Use the following syntax to reference a display key:

```
DisplayKey.get("NameOfTheKey")
```

Or if the key expects parameters:

```
DisplayKey.get("NameOfTheKey", Parameter1, Parameter2,... )
```



### Hint

View or modify a Display Key value

To view the value, just simply hold down the CTRL key and hover over the Display Key with your mouse. To modify the value, just simply hold down the CTRL key and left click on the Display Key with your mouse. In general, the same methods work the same way for other elements in Studio, such as entity fields, Gosu variables, PCF file references, etc.

**CTRL + Hover:** Tooltip

**CTRL + Click:** Open definition file



### Hint

Hover over for tooltips

Simply hover over the PCF element (or property) in the toolbox (or property window) to see the PCF documentation about what that element (or property) does.

## 5.4.1 Verification



### Activity

Verify the work you have done

#### 1. Reload the PCF changes

- In TrainingApp, reload the changes to the PCF file(s) and display keys.
- Verify your changes on the Summary page for William Andy.



### Keyboard shortcut

Reload PCF changes without restarting the Server

If you are running an open application project in Guidewire Studio and if internal tools are enabled, you can reload all the page configuration files and display keys for the server.

**Important:** If you reload PCF files while in edit mode, you may experience unpredictable results. For the current location, where there is a data modification in progress, the new PCFs may not be reloaded. Therefore, Guidewire recommends reloading PCF files while in read-only mode as it provides for more predictable results.

Use the ALT + SHIFT + L keystroke in the browser.

Person: William Andy

## Summary

Basics Social Media **Analysis**

ContactAnalysis

ContactTier <none>

CustomerRating

IsStrategicPartner ☐ Yes ☐ No

LastCourtesyContact MM/dd/yyyy

Fraud Investigations

WebAddress

## 5.5 Bonus Lab: Investigate optional widget properties

In this exercise, you will use the PCF Format Reference to investigate various widget properties.

### 1. Open the PCF Format Reference

- In Chrome, open the `pcf.html` file from the `C:\<installDirectory>\TrainingApp\modules\pcf.html`

### 2. Review the BooleanRadioInput widget properties

- Use the Quick Jump drop-down list to select the BooleanRadioInput widget.
- Review the widget properties.

### 3. Review the DateInput widget properties

- Use the Quick Jump drop-down list to select the DateInput widget.
- Review the widget properties.

### 4. Review the TextInput widget properties



- a) Use the Quick Jump drop-down list to select the TextInput widget.
- b) Review the widget properties.

#### 5. Review the TypeKeyInput widget properties

- a) Use the Quick Jump drop-down list to select the TypeKeyInput widget.
- b) Review the widget properties.

### 5.5.1 Write it down

**Requirement:** When a user attempts to modify the Is Verified widget value, a dialog box opens. The dialog box asks the user to confirm the change.

1. **Question: What is the name of the attribute (property) for a BooleanRadioInput that displays a confirmation message in a dialog box?**

**Requirement:** The label for the Evaluation Date must be in bold in the user interface.

2. **Question: What is the name of the attribute (property) for a DateInput that allows a configurator to implement a bold style for the label?**

**Requirement:** When a user hovers over a Legacy Code text input widget with their mouse cursor, the user interface displays, "The code must come from the legacy claim system."

3. **Question: What is the name of the attribute (property) for a TextInput that displays this message?**





## Solution

### 5.6 Lab Solution: Practice Dot notation

1. The creation date and time for anABContact.

```
anABContact.CreateTime
```

2. The tax status for anABContact.

```
anABContact.TaxStatus
```

3. The city for the primary address of anABContact.

```
anABContact.PrimaryAddress.City
```

4. All the buildings associated with anABContact.

```
anABContact.Buildings_Ext
```

5. Assuming that anABContact is a company, the number of employees for anABContact.

```
(anABContact as ABCompany).NumberOfEmployees
```

6. Assuming that anABContact is a legal venue, the type of court for anABContact.

```
(anABContact as ABLegalVenue).VenueType
```

## 5.7 Lab Solution: Determining widgets and their requirements



### Solution

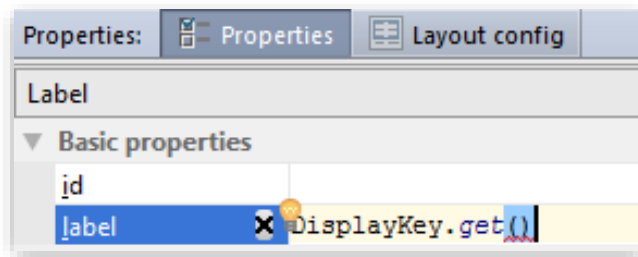
	Label/Info	Editable	Field	Datatype (Data Model)	Widget	valueType (Gosu)
1	Contact Analysis	n/a	n/a	n/a	Label	n/a
2	We need a horizontal divider	n/a	n/a	n/a	Input divider	n/a
3	Contact Tier	Yes	ContactTier	typekey.Contact Tier	TypeKey Input	typekey.Contact Tier
4	Customer Rating	Yes	CustomerRating_Ext	decimal	Text Input	java.math.BigDecimal
5	Strategic Partner	Yes	IsStrategicPartner_Ext	bit	Boolean Radio Button Input	java.lang.Boolean
6	Last Courtesy Contact	Yes	LastCourtesyContact_Ext	datetime	Date Input	java.util.Date
7	Fraud Investigation Number	No	FraudInvestigationNum_Ext	integer	Text Input	java.lang.Integer
8	Web Address	Yes	WebAddress_Ext	varchar(60)	Text Input	java.lang.String

## 5.8 Lab Solution: Add new atomic widgets

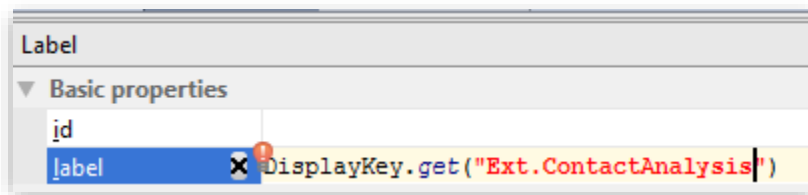


### Solution

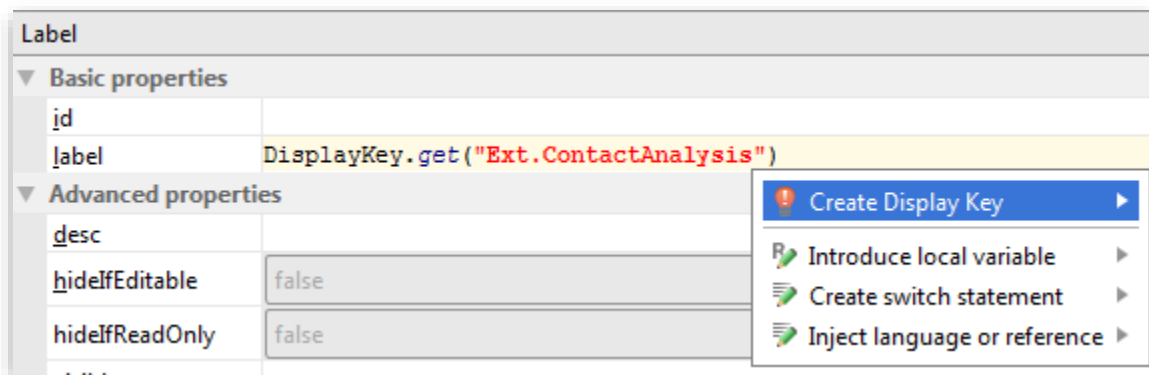
1. Call the `get()` function of the `DisplayKey` class in the label property.



2. Enter the name for the display key in the `get()` function as a String parameter (between double quotes).



3. Press **Alt + Enter**, and click **Create Display Key**:

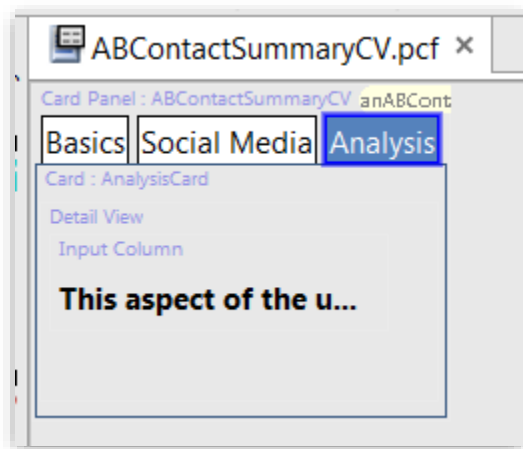


4. Then enter the text value for the locale and click **OK**



**5. Open the PCF file:**

- a) Open ABContactSummaryCV.pcf and select the Analysis card.



- b) To delete the existing label, right click on it and select Delete from the context menu.

**6. Drag and drop widgets inside the existing Input Column**

- a) Label widget for Contact Analysis label:

Label	
▼ Basic properties	
id	
label	DisplayKey.get("Ext.ContactAnalysis")

b) Input Divider widget to display a horizontal line:

Input Divider	
▼ Basic properties	
id	

c) TypeKey Input widget for Contact Tier

TypeKey Input	
▼ Basic properties	
editable	true
id*	ContactTier
label	DisplayKey.get("Ext.ContactTier")
required	
value*	anABContact.ContactTier
valueType*	typekey.ContactTier

## 7. Text Input widget for Customer Rating:

Properties:	Properties	Layout config	Reflection	PostOnChange
Text Input				
▼ Basic properties				
editable	true			
id*	CustomerRating			
label	DisplayKey.get("Ext.CustomerRating")			
required				
value*	anABContact.CustomerRating_Ext			

...and set the valueType to java.math.BigDecimal

validationExpression	
valueType	java.math.BigDecimal
visible	

#### 8. Boolean Radio Button Input widget for Strategic Partner:

Boolean Radio Button Input	
▼ Basic properties	
editable	true
falseLabel	
id*	IsStrategicPartner
label	DisplayKey.get("Ext.IsStrategicPartner")
required	
trueLabel	
value*	anABContact.IsStrategicPartner_Ext

#### 9. Date Input widget for Last Courtesy Contact:

Date Input	
▼ Basic properties	
dateFormat	<none selected>
editable	true
id*	LastCourtesyContact
label	DisplayKey.get("Ext.LastCourtesyContact")
required	
timeFormat	<none selected>
value*	anABContact.LastCourtesyContact_Ext

#### 10. Text Input widget for Number Of Fraud Investigations...

Text Input	
▼ Basic properties	
editable	false
id*	NumberOfFraudInvestigations
label	DisplayKey.get("Ext.NumOfFraudInv")
required	
value*	anABContact.FraudInvestigationNum_Ext

...and set the valueType to java.lang.Integer

validationExpression	
valueType	java.lang.Integer
visible	

#### 11. Text Input widget for Web Address:

Properties: Properties Layout config Reflection PostOnChange	
Text Input	
▼ Basic properties	
editable	true
id*	WebAddress
label	DisplayKey.get("Ext.WebAddress")
required	
value*	anABContact.WebAddress_Ext



The completed design:

The screenshot displays a web application window titled 'ABContactSummaryCV.pcf'. Inside, there's a 'Card Panel : ABContactSummaryCV' with three tabs: 'Basics', 'Social Media', and 'Analysis'. The 'Analysis' tab is selected, showing a 'Card : AnalysisCard' with a 'Detail View' and 'Input Column'. The main heading is 'Contact Analysis'. Below this, there are several input fields: 'Contact Tier' (a dropdown menu showing 'anABContact.ContactTier'), 'Customer Rating' (a text input field with 'anABContact.CustomerRati...'), 'Strategic Partner' (radio buttons for 'Yes' and 'No'), 'Last Courtesy Contact' (a date picker showing '././....'), 'Fraud Investigations' (a text input field with 'anABContact.FraudInvestig...'), and 'Web Address' (a text input field with 'anABContact.WebAddress\_...').

## 5.9 Bonus Lab Solution: Investigate optional widget properties



### Solution

**Requirement:** When a user attempts to modify the Is Verified widget value, a dialog box opens. The dialog box asks the user to confirm the change.

- Question:** What is the name of the attribute (property) for a BooleanRadiolInput that displays a confirmation message in a dialog box?

showConfirmMessage

**Requirement:** The label for the Evaluation Date must be in bold in the user interface.

2. **Question:** What is the name of the attribute (property) for a DateTimeInput that allows a configurator to implement a bold style for the label?

boldLabel

**Requirement:** When a user hovers over a Legacy Code text input widget with their mouse cursor, the user interface displays, "The code must come from the legacy claim system."

3. **Question:** What is the name of the attribute (property) for a TextInput that displays this message?

helpText