

Lesson 11 Code Generation and Debugging

We have just received the following bug report from the testers:

User Story: An insurance company wants to be able to "cascade" the ABCCompany's email address to all employees who do not have an individual email address.

Bug Description: This user story has already been implemented, but it is not working correctly. If I click on the 'Cascade Email Address' button then the application cascades the company's email address to all employees and not only to the ones that do not have an email address already. This results in overriding the person's email address which is not the desired behavior. Please fix it ASAP.

Priority: High

Screenshots:

The screenshot shows a software interface for managing company details. At the top, it says 'Company: Albertson's'. Below that is a 'Details' section with tabs for 'Company Info', 'Phone & Addresses', and 'Bank Accounts'. The 'Company Info' tab is selected. It contains fields for Name (Albertson's), Primary Contact (William Andy), Country (United States), Address 1 (345 Fir Lane), Address 2, Address 3, City (La Canada), County, and State (California). To the right of these fields is an 'Employee Info' section. It includes a checkbox for 'Can Have Employees?' which is checked, and a dropdown for 'Number of Employees' set to 4. Below this is a table titled 'Employees' with a 'Cascade Email Address' button. The table lists four employees: James Andersen (Email: jandersen@elegal.com), Samantha Andrews (Email: sandrews@andrewsmd.com), Eric Andy, and William Andy. At the bottom right is a 'Contact Insights' section.

Figure 1 Before clicking on Cascade Email Address

Company: Albertson's

Details

Company Info		Phone & Addresses	Bank Accounts
Name	*	Albertson's	
Primary Contact			
Primary Contact		William Andy	
Country	United States		
Address 1	345 Fir Lane		
Address 2			
Address 3			
City	La Canada		
County			
State	California		

Employee Info

Can Have Employees?

Number of Employees 4

Employees		Cascade Email Address
Name	Job Title	Email Address
James Andersen		info@Albertsons.com
Samantha Andrews		info@Albertsons.com
Eric Andy		info@Albertsons.com
William Andy		info@Albertsons.com

Contact Insights

Insight Score: 0

Figure 2 After clicking on Cascade Email Address

In this lab, you will debug a PCF method to identify a bug. Then, you will modify the PCF method and invoke the code generation. Finally, you will deploy the fixed PCF file and re-test the user story.

11.1 Prerequisites

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.

Verify that Albertson's has 4 employees as you can see it in Figure 1 above. Restore the development database if Albertson's has less than 4 employees or if all the employees have an email address.

11.2 Lab: Fix a bug

As a configuration developer, you want to be able to debug Guidewire applications. In this exercise you will use the debugger to debug a PCF method and identify an issue.

1. Open Guidewire Studio for TrainingApp

- a) From Studio, if your server is not already running, start the server using Debug 'Server'.
- b) Review the Debug console for errors and verify that the application is running in the Debug console.

2. Locate the function called by the Cascade Email Address button

- a) Log in to TrainingApp as Alice Applegate and go to the Details screen of Albertson's.
- b) Open the PCF file in Studio and identify the function specified in the action property of the Cascade Email Address button.

11.2.1 Lab: Write it down

1. Question: What is the name of the method called by the Cascade Email Address button?
2. Go to the function declaration and verify it is a PCF function. CTRL + Left click on the function name.
 - a) Verify that the Code tab opens in the PCF file



Guidewire API

The featured Guidewire API reference.

PCF methods are plain old Gosu functions defined in PCF files. The syntax is the same regardless where you define the function (PCF file, Enhancement or Gosu class). However, the semantics is slightly different: a PCF method...

- ...is local (private) in the PCF file that defines it (not even referenced files can access it)
- ...cannot be debugged at the place of definition (you have to open the generated <PCFFileName>Expressions.gs file after code generation, locate the method and put a breakpoint there)
- ...is called from a widget property (e.g.: action, value, available, editable, etc.)

PCF methods are defined on the Code tab of the main PCF Element – container or location – and have direct access to the root object(s).

Why would you write code in a PCF file as opposed to an Enhancement or Gosu class? Because the logic the function implements is tied to one or more widgets in that PCF file and it is not needed anywhere else. For example:

- determining the label (View or View/Edit) for a button
- determining if a TextInput widget is editable or not editable
- executing some initialization logic after navigating to a location

3. Open the appropriate Expressions.gs file and use the debugger to identify the issue

- a) Use a breakpoint, the Variables pane and 'Step over' button

4. Question: What is the issue? What solution would you propose?



Stop

11.2.2 Lab: Implement the fix



Reference

PCF file code generation

Remember, PCF files support both incremental and bulk code generation.

- Save the changed (CTRL + S) PCF file or select **Build** menu → **Make the project** to kick off incremental code generation.
- Select **CodeGen** menu → **Generate Page Configuration Classes** or **Build** menu → **Rebuild project** to kick off bulk code generation.

The code generation of a PCF file results in the following:

- The .gs file is a Gosu class that represents the PCF type.
- The .pcf is a binary file that describes the PCF file for the new PCF runtime.

1. Implement your proposed fix

- a) Fix the PCF method implementation in the ABContactDetailsCompanyDV.pcf.

Important: do not modify the Expressions.gs file.

- b) Kick off the code generation process by saving the file (CTRL + S).

2. Deploy the fix

- a) Verify that you can see your change in the ABContactDetailsCompanyDVExpressions.gs. Reload the PCF changes (ALT + SHIFT + L).

11.2.3 Verification



Activity

Verify the work you have done

1. Verify the issue has been resolved

- a) Log in to TrainingApp as Alice Applegate.
- b) Search for and open the Albertson's contact.
- c) For the contact, navigate to the Details page.
- d) Click Edit.
- e) Click on Cascade Email Address button.
- f) Verify that only contacts without an email address have been updated



11.3 Lab Solution: Fix a bug



1. Open Guidewire Studio for TrainingApp

A screenshot of the Guidewire Studio interface. The top menu bar shows "Debug" and "Server". The main window has tabs for "Debugger" (which is selected), "Console", and other tools. On the left, there's a toolbar with icons for file operations like open, save, and delete. The central area displays a terminal window with the following log output:

```
20:10:25,982 INFO MW-startThreadPool-15 poolSize=1
20:10:25,988 INFO Destination: 15 started
20:10:26,050 INFO Starting QPlexor...
20:10:26,053 INFO QPlexor started
20:10:26,135 INFO Increasing runlevel to 'MULTIUSER'
20:10:26,137 WARN
20:10:26,138 WARN !!! The server is in "development" mode.
20:10:26,138 WARN
20:10:26,138 WARN
20:10:26,138 WARN !!! The server is using the Dynamic Code I
20:10:26,138 WARN
20:10:26,143 INFO ***** ContactManager ready *****
20:10:26,155 INFO Started o.e.j.w.WebAppContext@7a67e3c6{/a}
```

At the bottom, there are tabs for "Debug" (selected), "TODO" (with 6 items), "Properties", "Terminal", and "Codegen".

2. Locate the function called by the Cascade Email Address button

The screenshot shows the PCF editor interface with two tabs at the top: 'ABCompanyDetailsPage.pcf' and 'ABContactDetailsCompanyDV.pcf'. The main area displays a form with various input fields and sections. A toolbar button is selected, highlighted with a blue border. The 'Properties' tab is open, showing the following details for the toolbar button:

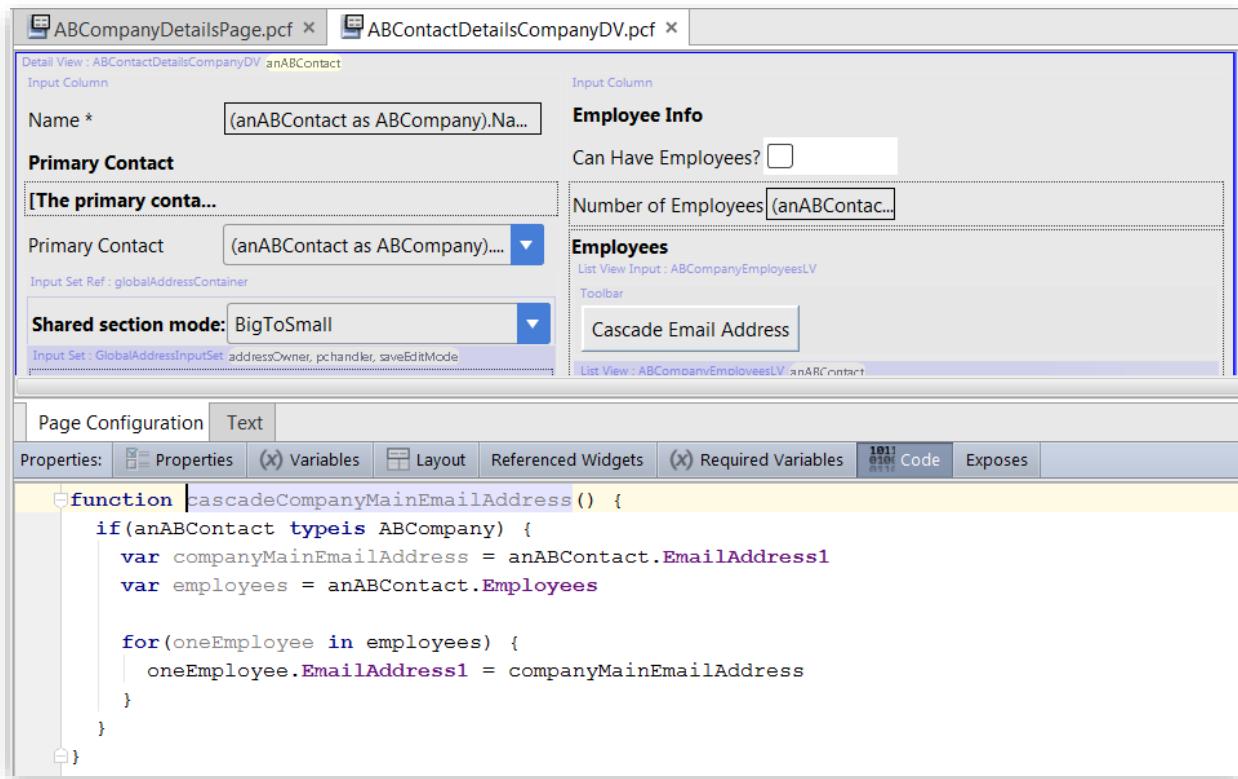
Property	Value
action	cascadeCompanyMainEmailAddress()
id*	CascadeEmailAddress
label	DisplayKey.get("Training.Company.CascadeEmailAddress")

11.3.1 Lab Solution: Write it down

1. Question: What is the name of the method called by the Cascade Email Address button?

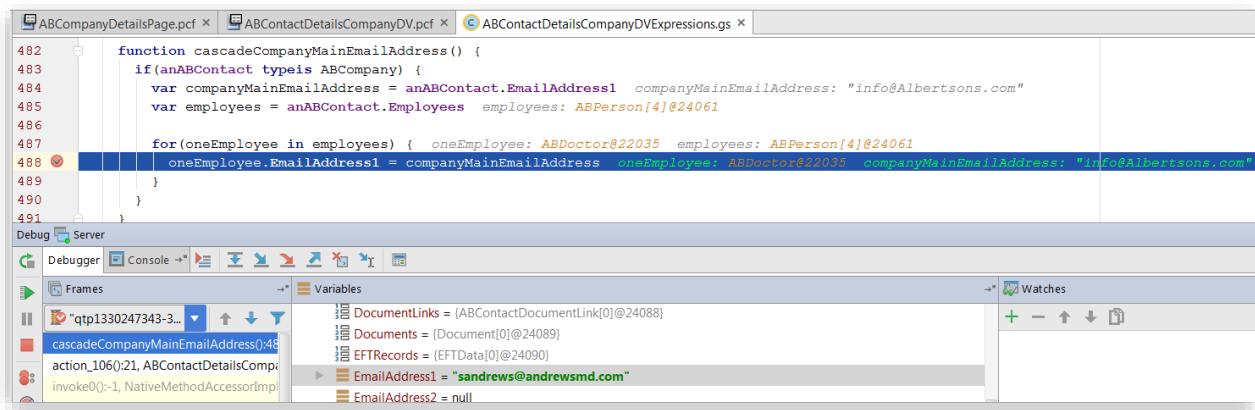
```
cascadeCompanyMainEmailAddress()
```

2. Go to the function declaration and verify it is a PCF function



3. Open the appropriate Expressions.gs file and use the debugger to identify the issue

- a) Use a breakpoint, the Variables pane and 'Step over' button



4. Question: What is the issue? What solution would you propose?

The highlighted line in the above screenshot will be executed, even though we can see that the EmailAddress1 field of the oneEmployee is already set to sandrews@andrewsmd.com. Suggested solution: put the line in an if statement and execute it only if the oneEmployee.EmailAddress1 is null.

11.3.2 Lab Solution: Implement the fix



1. Implement your proposed fix

The screenshot shows the Oracle ADF Faces page editor with the 'ABCompanyDetailsPage.pcf' page open. The page contains a form with various input fields and sections. In the 'Code' tab of the properties panel at the bottom, the following JavaScript code is visible:

```
function cascadeCompanyMainEmailAddress() {
    if(anABContact typeis ABCCompany) {
        var companyMainEmailAddress = anABContact.EmailAddress1
        var employees = anABContact.Employees

        for(oneEmployee in employees) {
            if(oneEmployee.EmailAddress1 == null) {
                oneEmployee.EmailAddress1 = companyMainEmailAddress
            }
        }
    }
}
```

You also have the option to copy and paste it from below. However, it is recommended that you type the code in.

```
function cascadeCompanyMainEmailAddress() {
```

```

if(anABContact typeis ABCompany) {
    var companyMainEmailAddress = anABContact.EmailAddress1
    var employees = anABContact.Employees

    for(oneEmployee in employees) {
        if(oneEmployee.EmailAddress1 == null) {
            oneEmployee.EmailAddress1 = companyMainEmailAddress
        }
    }
}
}

```

2. Deploy the fix



The screenshot shows a code editor window with three tabs at the top: "ABCompanyDetailsPage.pcf", "ABContactDetailsCompanyDV.pcf", and "ABContactDetailsCompanyDVExpressions.gs". The "ABContactDetailsCompanyDVExpressions.gs" tab is active, displaying the following Java script code:

```

property set anABContact ($arg : ABContact) { setRequireValue("anABContact", 0,
    function cascadeCompanyMainEmailAddress() {
        if(anABContact typeis ABCompany) {
            var companyMainEmailAddress = anABContact.EmailAddress1
            var employees = anABContact.Employees

            for(oneEmployee in employees) {
                if(oneEmployee.EmailAddress1 == null) {
                    oneEmployee.EmailAddress1 = companyMainEmailAddress
                }
            }
        }
    }
}

```

A yellow warning icon is visible next to the line of code where the variable 'oneEmployee' is used without being declared.