Solution: Business Rules

In this lab, you will write and verify a preupdate rule for contacts of the type ABDoctor that do not have a defined doctor specialty.

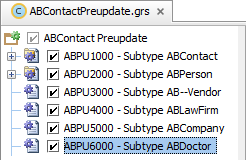
Requirements

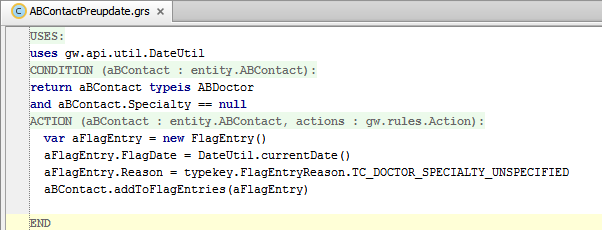
This lab requires that you use TrainingApp 8.0, Guidewire Studio 8.0, and a supported web browser. To view, edit, and delete various contacts, log in to TrainingApp as Super User. The default URL for TrainingApp is <http://localhost:8880/ab/ContactManager.do>. The login/password for Super User is su/gw.

1. Create a ABContact preupdate rule

In this exercise, you will first create a preupdate rule that creates a flag entry for a contact of the type ABDoctor that does not have a defined doctor specialty. Next, you will debug your rule using the debugger. Then, you will verify the execution of your rule.

Tasks

1. Open Guidewire Studio for TrainingApp
2. Create an ABContact preupdate rule for contacts of the subtype ABDoctor that creates a flag entry when a contact of the type ABDoctor does not have a defined specialty
3. Create the new rule using a naming convention that reflects its purpose and execution order.  
   
4. In the Rule Set Editor, write Gosu code that meets the following condition criteria:
   * The contact is of the type ABDoctor
   * The Specialty field is null
5. In the Rule Set Editor, write Gosu code that performs the following actions:
   * Creates a new flag entry
   * Specifies the flag date as the current date
   * Specifies the flag entry reason as an unspecified specialty for the doctor
   * Adds the flag entry to the flag entries array for the given contact



USES:

uses gw.api.util.DateUtil

CONDITION:

return aBContact typeis ABDoctor

and aBContact.Specialty == null

ACTION:

var aFlagEntry = new FlagEntry()

aFlagEntry.FlagDate = DateUtil.currentDate()

aFlagEntry.Reason = typekey.FlagEntryReason.TC\_DOCTOR\_SPECIALTY\_UNSPECIFIED

aBContact.addToFlagEntries(aFlagEntry)

1. Deploy the rule

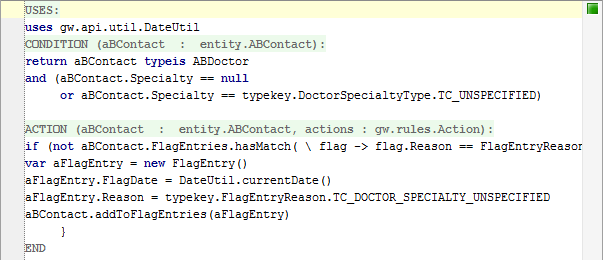
Verification

1. Set break points
2. In the Rule Set Editor, set a breakpoint in the rule action that creates a new flag entry.
3. Log in to TrainingApp
4. From Studio, if your server is not already running, start the server using Debug 'Server'.
5. Review the Debug console for errors and verify that the application is running in the Debug console.
6. Log in as Super User.
7. Change the doctor specialty for Rebecca Stevens
8. Open the Rebecca Stevens contact.
9. In the Details page, in the Person Info tab, in the Medical Specialty input set, change the Specialty to <none>.
10. Click Update.
11. Verify that Studio hit the break point
12. In Studio, view the hit point in the Debugger.
13. Review the variables.
14. Resume the program.
15. Edit the flag entry
16. In TrainingApp, view the summary page for Rebecca Stevens.
17. In the Flag Entries list view, for the row with a warning flag and the "Doctor Specialty is unspecified" reason, click View/Edit.
18. In the Flag Entry popup, in the Resolution field, enter your name.
19. Click Update.
20. Verify that the flag entry is no longer flagged.
21. Modify your ABContact preupdate rule

In this exercise, you will modify your rule so that the rule only creates one flag entry for a contact of the type ABDoctor that does not have a defined doctor specialty. You will then debug your rule using the debugger and verify the execution of your rule.

Tasks

1. Modify the ABContact preupdate rule for contacts of the subtype ABDoctor that creates a flag entry when a contact of the type ABDoctor does not have a defined specialty
2. In the Rule Set Editor, modify the condition for your rule to meets the following condition criteria:
   * The contact is of the type ABDoctor
   * The Specialty field is null or is unspecified
   * There is not already an open note for the contact with a flag reason of an unspecified specialty



USES:

uses gw.api.util.DateUtil

CONDITION:

return aBContact typeis ABDoctor

and (aBContact.Specialty == null or aBContact.Specialty == typekey.DoctorSpecialtyType.TC\_UNSPECIFIED)

ACTION:

if (not aBContact.FlagEntries.hasMatch( \ flag -> flag.Reason == FlagEntryReason.TC\_DOCTOR\_SPECIALTY\_UNSPECIFIED and flag.IsOpen)) {

var aFlagEntry = new FlagEntry()

aFlagEntry.FlagDate = DateUtil.currentDate()

aFlagEntry.Reason = typekey.FlagEntryReason.TC\_DOCTOR\_SPECIALTY\_UNSPECIFIED

aBContact.addToFlagEntries(aFlagEntry)

}

1. Deploy the modified rule

Verification

1. Set break points
2. In the Rule Set Editor, set a breakpoint in the rule action that creates a new flag entry.
3. Log in to TrainingApp
4. From Studio, if your server is not already running, start the server using Debug 'Server'.
5. Review the Debug console for errors and verify that the application is running in the Debug console.
6. Log in as Super User (su/gw).
7. Change the doctor specialty for Rebecca Stevens
8. Open the Rebecca Stevens contact.
9. In the Details page, in the Person Info tab, in the Medical Specialty input set, change the Specialty to Unspecified.
10. Click Update.
11. Verify that Studio hit the break point
12. In Studio, view the hit point in the Debugger.
13. Review the variables.
14. Resume the program.
15. Verify the flag entry and rule behavior
16. In TrainingApp, view the summary page for Rebecca Stevens.
17. In the Flag Entries list view, for the row with a warning flag and the "Doctor Specialty is unspecified" reason, click View/Edit.
18. In the Flag Entry popup, click Cancel.
19. In the Details page, in the Person Info tab, in the Medical Specialty input set, change the Specialty from Unspecified to <none>.
20. Click Update.
21. You should NOT hit the break point in the rule.
22. View the summary page for Rebecca Stevens.
23. In the Flag Entries list view, verify that there is only one row with a warning flag and the "Doctor Specialty is unspecified" reason.
24. Remove the break points
25. In Studio, remove all breakpoints.

|  |  |
| --- | --- |
|  | Stop and ask your instructor to review your completed lab. |