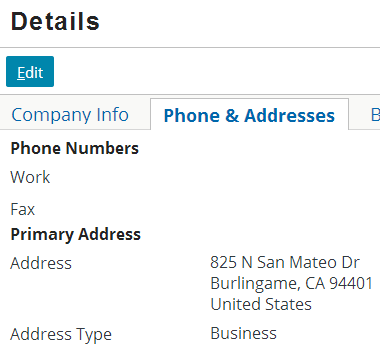
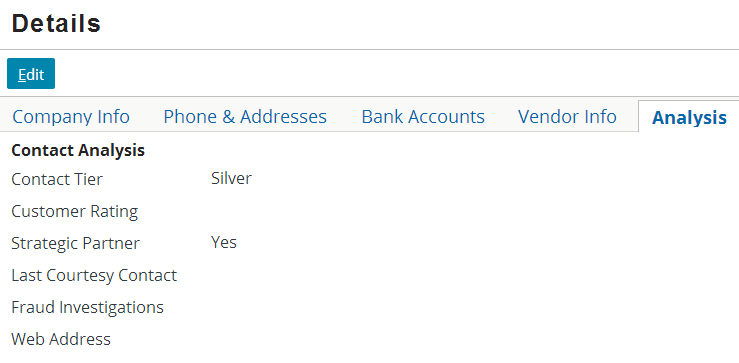
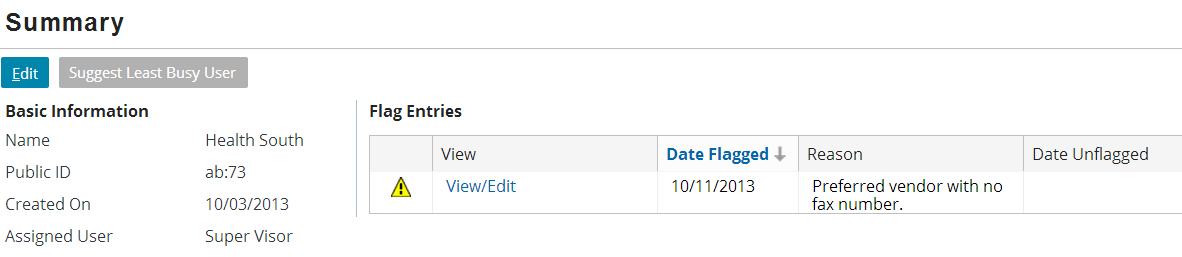
Lab: Business Rules

In this lab, you will write pre-update rules that flag specific entries.

Overview

In TrainingApp, an ABCompany contact that is a strategic partner should have a fax number. There is an ABContact.IsFlagged property, a flag icon in the info bar that displays only if the contact is flagged, and an array of flag entries where an entry has a FlaggedReason field.



Tasks

1. Write an ABContact pre-update rule that:

* Executes only for an ABContact that is a strategic partner and a company (or any of its subtypes) where the fax number is NULL
* Does nothing if the contact is already flagged for this reason
* Otherwise creates a new flag entry instance where the Reason field is set to "vendor\_fax" and the flag date field to the current date and adds the new flag entry instance to the contact's FlagEntries array (HINT: There is an existing rule that flags ABPerson contacts that have no email address. You can use this as a guide for programmatically adding a new flag entry)

1. To test your rule, find a company that has no fax number (such as Burlingame Saab) and modify it so that it is a strategic partner. Confirm that the new flag entry is created.

Verification

1. Practice debugging the rule by setting a breakpoint in the condition for the first pre-update rule and running with in debug server mode. Pick another company with no fax number (such as Health South) and change it to be a strategic partner. Step through the code to verify that the pre-update rules are executed as expected.

|  |  |
| --- | --- |
|  | Stop and ask your instructor to review your completed lab or compare your solution with the provided solution set. |

abcontactdetailsCompanyDV.pcf

package rules.Preupdate.ABContactPreupdate\_dir

uses gw.api.util.DateUtil

@gw.rules.RuleName("rumneek test")

internal class rumneektest {

static function doCondition(aBContact : entity.ABContact) : boolean {

/\*start00rule\*/

return aBContact.IsStrategicPartner\_Ext

and

(aBContact typeis ABCompany)

and (aBContact.FaxPhone==null || aBContact.FaxPhone.Empty )

/\*end00rule\*/

}

static function doAction(aBContact : entity.ABContact, actions : gw.rules.Action) {

/\*start00rule\*/

if (not aBContact.FlagEntries.hasMatch(\ f -> (f.Reason == FlagEntryReason.TC\_VENDOR\_FAX) and (f.UnflagDate == null))) {

var newEntry = new FlagEntry()

newEntry.FlagDate = DateUtil .currentDate()

newEntry.Reason=FlagEntryReason.TC\_VENDOR\_FAX

print("RUMNEEK \*\*\*\*\*\*\*\*\*\*\*\*\*")

aBContact.addToFlagEntries(newEntry)

}

/\*end00rule\*/

}

}