Lesson 2 Batch Processes

This lab requires that you use **TrainingApp**, **Guidewire Studio**, and a supported web browser. Start **Guidewire Studio** for **TrainingApp**. Start the server as Debug 'Server'.

The default URL for **TrainingApp** is: `http://localhost:8880/ab/ContactManager.do`. Log in to **TrainingApp** as Super User whose login/password is **su/gw**.

2.1 Create a custom batch process

Succeed Insurance needs a batch process that flags a contact if they are missing an assigned business user.

2.1.1 Requirements

Spec 1 Create a batch process that creates a flag entry when a contact does not have a user assigned to them.

Spec 2 If an identified contact already has an open flag entry for this reason, the batch process does not create an additional flag entry.

Spec 3 Create a custom FlagEntryReason typecode to support this use case:

- code = contactnoassigneduser
- desc = Contact missing assigned user.
- name = Contact missing assigned user.

Spec 4 Spec 4 Create a custom BatchProcessType typecode to support this use case:

- code = contactnoassigneduser
- desc = Contact missing assigned user.
- name = Contact missing assigned user.

Spec 5 Must be schedulable and executable from the UI batch process screen.

Spec 6 Must be able to stop the batch process during processing.

Spec 7 The solution should consider that thousands of contacts could conceivably be added during a single day. It is critical that this process executes successfully without consuming system resources unnecessarily. Re-runs in this environment are particularly costly and inconvenient.

Spec 8 Batch process package called `batch.flagcontactnoassigneduser`.

Spec 9 Batch process class called `FlagContactNoAssignedUserBatch`.

2.1.2 Tasks

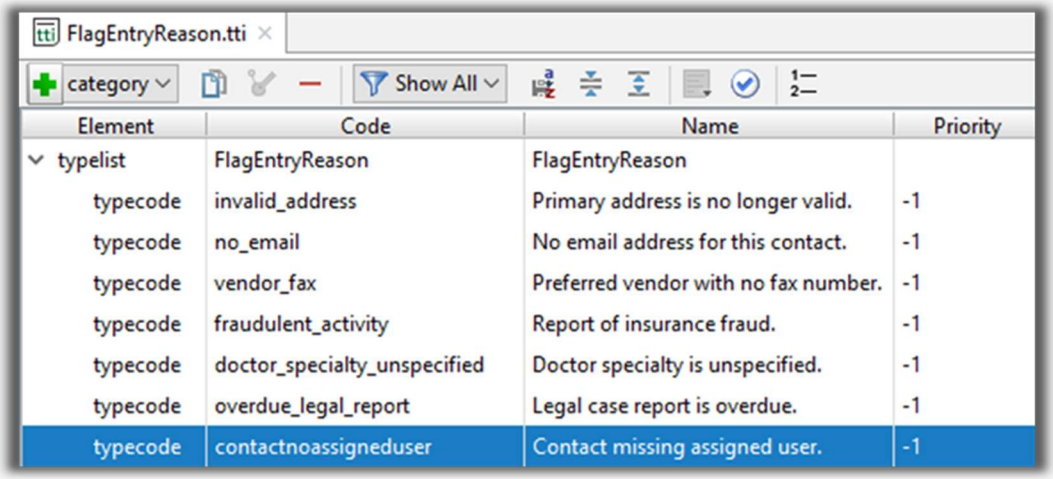
1. **Create new typecode in FlagEntryReason typelist.**
2. **Create new typecode in BatchProcessType typelist and add categories.**
3. **Create new batch process package.**
4. **Create batch process class that implements the specifications.**
5. **Modify the ProcessesPlugin plugin found under trn.ta.batch.ProcessPlugin.**
6. **Deploy code changes.**
7. **Perform verification steps.**

2.1.3 Verification steps

1. **Login into TrainingApp.**
2. **Setup test contact cases by:**
   a) Setting the assigned user field for one contact
   b) Not setting the assigned user field for the other contact
3. **Run the batch process manually.**
   a) **ALT + SHIFT + T**.
   b) Click the **Run** button.
4. **Verify expected results.**
   a) Find a contact where the assigned user is not set.
   b) A flag entry should exist.
5. **Run the batch process again.**
   a) Verify expected results.
   b) No duplicate flags should be set.

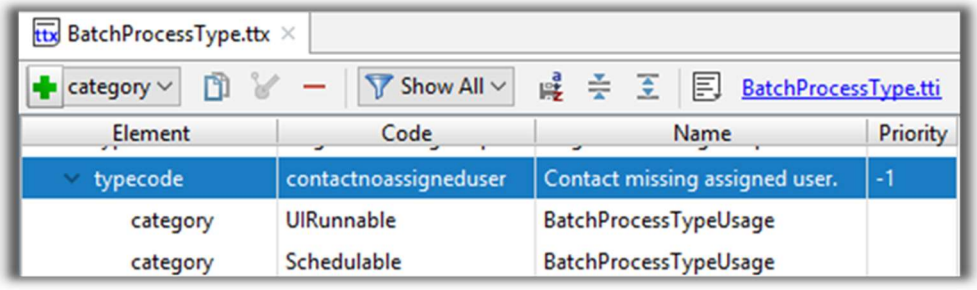
2.1.4 Solution

1. **Create new typecode in FlagEntryReason typelist.**
   a) Open **FlagEntryReason** typelist.
   a) Add new typecode.



Element	Code	Name	Priority
typelist	FlagEntryReason	FlagEntryReason	
typecode	invalid_address	Primary address is no longer valid.	-1
typecode	no_email	No email address for this contact.	-1
typecode	vendor_fax	Preferred vendor with no fax number.	-1
typecode	fraudulent_activity	Report of insurance fraud.	-1
typecode	doctor_specialty_unspecified	Doctor specialty is unspecified.	-1
typecode	overdue_legal_report	Legal case report is overdue.	-1
typecode	contactnoassigneduser	Contact missing assigned user.	-1

2. **Create new typecode in BatchProcessType typelist and add categories.**
   a) Open BatchProcessType typelist.
   b) Add new typecode.



Element	Code	Name	Priority
typecode	contactnoassigneduser	Contact missing assigned user.	-1
category	UIRunnable	BatchProcessTypeUsage	
category	Schedulable	BatchProcessTypeUsage	

3. **Create new batch process package.**
   a) Right-click on **si.ta** package and select **New → Package**.
   b) Enter **batch.flagcontactnoassigneduser** as the new package name.

4. **Create batch process class that implements the specifications.**
   a) Right-click on **flagcontactnoassigneduser** package and select **New → Gosu Class**.
   b) Enter `FlagContactNoAssignedUserBatch` as the new class name.
      - Extend the BatchProcessBase class.
      - Add constructor.
      - Configure the **doWork** method.

```
package si.ta.batch.flagcontactnoassigneduser

uses com.google.common.collect.Iterables
uses gw.api.database.Query
uses gw.api.database.Relop
uses gw.api.util.DateUtil
uses gw.processes.BatchProcessBase
uses gw.transaction.Transaction

class FlagContactNoAssignedUserBatch extends BatchProcessBase {

  construct() {
    super(BatchProcessType.TC_CONTACTNOASSIGNEDUSER);
  }

  override function requestTermination() : boolean {
    super.requestTermination()
    // updates the built-in TerminateRequested flag to true
    // to indicate a termination request will be evaluated in doWork()
    return true
  }

  override function doWork() : void {
    // construct the main query selecting all contacts with no assigned user
    var mainQry = Query.make(ABContact)
    mainQry.compare(ABContact#AssignedUser, Relop.Equals, null)

    // construct a subquery finding all open FlagEntries for a given Reason
    var subQry = Query.make(FlagEntry)
    subQry.compare(FlagEntry#IsOpen, Equals, true)
    subQry.compare(FlagEntry#Reason, Equals, FlagEntryReason.TC_CONTACTNOASSIGNEDUSER)

    // join the main and sub queries via subselect to only retrieve eligible persons
    var results = mainQry.subselect(ABContact#ID, CompareNotIn, subQry, FlagEntry#ABContact).select()
    this.OperationsExpected = results.Count

    //feed the result set in manageable chunks
    this.setChunkingById(results, 1000)

    //split each chunk into smaller bundle-sized partitions
    var partitions = Iterables.partition(results, 100)

    for (eachPartition in partitions) {
      // check the built-in TerminateRequested flag at the start of each partition
      if (this.TerminateRequested) {
        break
```

```
      } else {
        Transaction.runWithNewBundle(\trn -> {
          for (eachContact in eachPartition) {
            // check the built-in TerminateRequested flag for each person iteration
            if (this.TerminateRequested) {
              break
            }
            // -- the app server side FlagEntries array filtering would also work
            // -- but very inefficient compared to the database subselect used above
            //if (!eachPerson.FlagEntries.hasMatch(\currentEntry ->
            //    currentEntry.IsOpen and currentEntry.Reason == FlagEntryReason.TC_NOEMAILORPHONE)) {
            try {
              var newFlagEntry = new FlagEntry()
              newFlagEntry.Reason = FlagEntryReason.TC_CONTACTNOASSIGNEDUSER
              newFlagEntry.FlagDate = DateUtil.currentDate()
              eachContact.addToFlagEntries(newFlagEntry)
            }
            catch (e : Exception) {
              this.incrementOperationsFailed()
              throw e
            }
            finally {
              // Counts "completed" not "successful": Completed - Failed = Sucessful.
              this.incrementOperationsCompleted()
            }
            //}
          }
        }, "su")
        // identifies the user to use when committing changes, required
        // for batch processes that are scheduled and have no inherent user
      }
    }
  }
}
```

5. Modify the ProcessesPlugin plugin found under trn.ta.batch.ProcessPlugin.

```
package trn.ta.batch

uses gw.plugin.processing.IProcessesPlugin
uses gw.processes.BatchProcess
uses si.ta.batch.flagcontactnoassigneduser.FlagContactNoAssignedUserBatch
uses trn.ta.batch.legalreport.FlagOverdueLegalReportsBatch

class ProcessesPlugin implements IProcessesPlugin {

  override function createBatchProcess(type : BatchProcessType, arguments : Object[]) : BatchProcess {
    switch(type) {
      case BatchProcessType.TC_FLAGOVERDUELEGALREPORTS:
        return new FlagOverdueLegalReportsBatch()
      case BatchProcessType.TC_CONTACTNOASSIGNEDUSER:
        return new FlagContactNoAssignedUserBatch()
      default:
        return null
    }
  }
}
```

6. **Deploy code changes.**
   c) From the Studio menu, **Restart** the server.
7. **Perform verification steps.**